N-Link Modular Smart Robotic Arm

A Major Qualifying Project
submitted to the Faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree in Bachelor of Science

Submitted by:
Ryan Eastwood [RBE/CS]
Tim Fromme [RBE/CS]
Tej Sheth [RBE/ME]
Nicholas Weddington [RBE/ECE]

Project Advisors:
Cagdas D. Onal PhD, Advisor [RBE/ME]
Loris Fichera PhD, Co-advisor [RBE/CS]
Jie Fu PhD, Co-advisor [RBE/ECE]

Abstract

Today, 2.6 million industrial robots have been deployed for applications worldwide, allowing us to produce, sort, and create products faster. Often, these robots require maintenance every 3,850 hours and have downtimes that yield production resulting in loss of efficiency and increase the cost. The goal of this MQP is to develop a modular robotic arm with n-links that can position its end effector in a 3D task space regarding of the number of links attached. It is also a priority that the arm is fault-tolerant and can be removed and attached in real-time without requiring a reboot. Creating this arm demonstrated how a modular robotic arm can operate in a dynamic task space by adding various links and decrease malfunction maintenance time with the improvement of fault tolerance. The results of this project show a detailed evaluation for the modular robotic arm that verifies that the arm would operate as predicted if it were fully constructed.

Acknowledgements

We would like to thank our project advisor Professor Cagdas Onal, and co-advisors Professors Jie Fu and Loris Fichera for all their support and guidance for our project throughout the year.

Contents

Figures

Introduction

Robotic arms today are used in many factories worldwide, performing tasks at a high rate as part of an assembly line. Failure of these arms can stall the whole line, leading to large amounts of lost efficiency, costing the factory time and money. Developing arms that can perform even with broken equipment can allow these assembly lines to operate with no downtime at all, outside of short maintenance periods. Redundant arms can be added at a sizable cost to act in failure conditions but establishing redundancy within an arm unit would allow for cheaper and more compact improvements to reliability. Modular robotics offers an interesting solution to adding redundancy. By creating a system capable of easily being reconfigured, faulty joints in an arm can be treated as static links (i.e. simple transformations). As the joints fail the robot loses access to the outer reaches of its task space, however, given a starting configuration containing redundancy, the arm will retain most of its manipulability in the case of a single link failure. If the system allows for quick maintenance, failures can be minimized to mostly incidents of single link failure. To attain a fast and easy maintenance, the arm is designed to be modular in nature, such that faulty links can be removed and replaced in a matter of seconds.

In this project we developed an N-link modular robotic arm that is fault tolerant, giving the robot the ability to operate even when minor malfunctions occur. The robot has have reconfigurable joints so that when one of the joints fail, it can be replaced, or if additional links need to be added or subtracted to complete a task within a specific space, that can be done to change the distance that the arm can reach. This platform would allow a single link to be manufactured in mass quantities to serve a wide variety of configurable applications. Modular robots accommodate for an extensive amount of task, so implementing more of them in industry will resemble a single solution that improves production and usage of the arm. Our final design was simulated as the project deviated from the original plan to construct a physical prototype due to complications of the COVID-19 pandemic.

Goal Statement

The goal of this project is the development of an n-link robotic arm that is capable of dynamically positioning its end effector at a given pose in 3-space depending on the number of modules attached. The system was able to handle adding or removing links in real-time with no reboot required. The system can also detect failure states in links and modify its calculations to account for them and keep the end effector going to the correct pose. Possible failure states include a motor dying, a microcontroller not responding or dying, and software errors. As long as the communication line remains up and a minimum of four links are attached and functional, the robot should be able to reach its desired pose.

The system requires a minimum of four links and base link to guarantee its ability to reach the desired pose because a minimum of four degrees of freedom are required for an arbitrary pose to be reachable within the task space. The task space was restricted to the full task space of the first three links in the system. If the task space were any larger, there would be no guarantee of being able to reach every pose in the task space if some of the links failed. In addition, we assumed one successful boot of the system without any failing links to be able to accurately save the configuration and properly model it in software.

Project Distribution

Although this project is interdisciplinary, all four team members took a lead in a specialized area of engineering and development:

Ryan Eastwood [RBE/CS] - Implemented kinematics algorithms and simulation

Tim Fromme [RBE/CS] - Designed and implemented the low-level communication protocols

Tej Sheth [RBE/ME] - Designed and conceptualized all mechanical components and assemblies

Nicholas Weddington [RBE/ECE] - Designed and implemented all electrical design and integration

Background

Fault Tolerance

For a robot to achieve fault tolerance behavior, the mechanical system must be capable of avoiding damage due to unexpected stresses or strains (Zhou, Liu, Wang, & Xu, 2018). The applications of mechanical fault tolerance design vary immensely. In some cases, it is beneficial to make the joint limp and free moving to prevent damage to the joint gearbox, motor, etc. In other cases, it is far better to immediately lock the joint in place to avoid irreversible damage to the mechanism. The advantages and drawbacks of these design considerations depend largely on the expected behavior and use cases of the electromechanical system.

One of the features of the modular link concept is to accommodate failure in neighboring links. In the event of mechanical failure, this can be achieved by immediately locking the joint in its current position and providing feedback. For example, if a motor fails, the joint is expected to hold its position and send positional feedback so that the system can dynamically account for the error. This error can be accounted in real time by adjusting the position of the neighboring joints.

The joint locking mechanism may be passive or active. A passive joint locking mechanism would not require communication from the system to maintain the joint’s position. In contrast, an active joint locking mechanism would need to be notified of a mechanical error to lock the joint. Both options will be explored.

Kinematics Algorithms

The calculations that form the relation between joint angles and task space position are referred to as kinematics. Forward kinematics is the calculation of task space given joint angles. In forward kinematics, a transformation matrix representing the transform from the base frame to the end effector is solved for current joint angles, yielding a matrix containing end effector orientation and positions. Inverse kinematics seeks to do the opposite, supplying an end effector position, and receiving the joint angles necessary to reach that position. While every joint angle configuration corresponds to one end effector position, end effectors can relate to multiple joint configurations. Because of these multiple solutions, inverse kinematics faces increased difficulty in calculations, with a method for resolving multiple solutions needing to be added. This issue becomes more apparent with a redundant system, or a kinematic chain that has more than the necessary number of joints to achieve all 6 degrees of motion at the end effector. In redundant systems, the basic inverse calculation becomes unsolvable, with every end effector position containing multiple solution sets of joint angles. Redundant arms offer the benefit of manipulating the position of intermediate links to avoid obstacles and offer some failure resistance. If a redundant joint fails, the robot still possesses 6 degrees of freedom. The scope of our project focuses on redundancy as a fault tolerance feature, so our kinematics must solve an unconstrained problem with numerous solutions.

In the solving of inverse kinematics, many methods exist, typically using the Jacobian and its variations. Raw inverse calculations may not always have a singular closed form solution, so the Jacobian methods are used, causing the end effector to converge on the desired location instead. Jacobian solutions typically operate in an iterative manner. Solving a Jacobian method will yield a change in angle, rather than a goal angle. By updating the current angle with the new change and recalculating the solution, Jacobian inverse methods converge on potential solutions.

The Jacobian is not always invertible, especially when approaching singularities where it will lose rank. One alternative method is to use the transpose of the Jacobian rather than the inverse in the form $\Delta\theta = \alpha J^T e$ for the appropriate scalar value alpha, where 0< α <1. While the transpose is not equivalent in any way, the $\Delta\theta$ produced by the equation can be proven to reduce the error vector, allowing for convergence.

The Damped least squares method uses a minimization technique to cause convergence to a point. DLS excels at returning stable deltas and can be optimized even further through careful selection of a scaling factor λ. Damped least square seeks to minimize the output of the equation

$$\|J * \Delta\theta - e\|^2 + \lambda^2\|\Delta\theta\|^2$$

This can be modeled as:

$$\Delta\theta = J^T(JJ^T + \lambda^2 * I)^{-1} * e$$

Solving this equation yields a set of theta values that when added to the current joint angles will converge towards a desired point. It is possible that these theta values will overshoot the set point. In this equation e is the error vector given by the distance in task space from end effector to desired position. The DLS method allows the arm to perform properly near singular configurations at the cost of a loss in accuracy caused by the damping factor. To reduce the loss in accuracy while still benefitting from the singularity protection, the damping coefficient lambda can by dynamically modified in response to the proximity to singular configurations. (Caccavale, 1997) By taking the singular value decomposition of the Jacobian, the matrix of singular values can be obtained. The minimum value of this matrix approaches zero as the arm nears a singular configuration. Using this minimum singular value, the damping factor can scale to be higher when needed and reduced when it is unnecessary.

Communication Protocols

SPI

Serial Peripheral Interface (SPI) is a standard for serial communication between devices. It is a One Master to Many Slave architecture. It can transmit in full duplex and requires a minimum of four lines. There is a Clock line that determines the communication speed, with one bit of data per rising edge of the clock. The two data lines are MOSI and MISO being Maser-Out-Slave-In and Master-In-Slave-Out. The last lines are the Slave Select lines, of which there is one per slave device. The Slave Select is active low, so they are held high by the master until it needs to communicate with a slave. The master will pull one Slave Select line low and communicate with that one slave over the MOSI and MISO lines. (Dhaker, 2018)

I2C

Inter-Integrated Circuit (I2C) is another standard for serial communication between devices. It does not require differentiation between Master and Slave. There are only two lines, the Clock line that determines communication speed, and the Data line. Each node has a fixed address used to address it. To communicate, one device will send the Start Condition, followed by the target address, followed by the data it wants to send, ending with the Stop Condition. By using addresses, it only requires 2 lines total for any number of connected devices, however it only supports half-duplex communication. (I2C Bus, Interface, and Protocol, 2020)

CAN

Controller Area Network (CAN) is a third standard for serial communication between devices. It does not require differentiation between Master and Slave because each message is a broadcast message to all other nodes. There are only two lines, CANH and CANL, which together determine if the bit is a 0 or 1. If both lines are at half-voltage, it is called Recessive and signals a logic 0. If CANH is pulled high and CANL is pulled low, it is called Dominant and signals a logic 1. Each message contains a message ID which is also the priority of the message. The CAN bus is architected in such a way that lower message IDs will win a process called Arbitration, which will keep that message on the line in the case that two nodes try to send data at the same time. The losing node will try to send its message again later. All nodes listen to all messages since it is a broadcast network. The CAN protocol also specifies error-detection protocols and flow control in the case of traffic congestion. (Corrigan, August 2002 - Revised May 2016)

Power Architecture

Power architecture is important when deciding the power requirements for each of the components in a modular robotic system. As the number of loads increased, distributed DC power architectures were developed to supply power to the loads through a number of perfect electric conductors (PECs). (Prabhala V. A., 2018) These different PECs provide different voltage level outputs catering to various loads and enables the use of standardized low power modules that make it easy to customize an existing system for any special requirements in the future. (Prabhala V. A., 2018) Implementing good power architecture is critical to operating a fully functional modular robotic arm because the input voltage varies as multiple components are connected.

A voltage regulator is used to provide a constant DC voltage between its output terminals, allowing the output voltage to remain constant despite changes in the load current drawn from the output terminal and changes in the power supply that feeds the regulator. (Sedra, 1982) Voltage regulators can be installed along distribution lines so that certain components that require a specific voltage independent of the larger amount of power drawn from the main voltage supply line. The distributed DC power architecture separates the power lines which is important because they can generate a lot of heat and possibly blow out the system. (Korinek, 2009)

Sensors

Current Sensing

Current sensor monitoring systems is increasingly demanded by electronic applications for monitoring purposes and digital control help improve efficiency and reduce loss while protecting the system from itself and its surroundings. By measuring the current that is drawn from the motor, the torque on the motor and each joint can be calculated. (Giovino, 2010) Current sensing can be divided into four techniques: Ohm's law of resistance, Faraday's law of induction, magnetic field sensors, and faraday effect. (Ziegler, Woodward, Iu, & Borle, 2009) Most current sensors use low-value resistors and measure the current through it using a voltage drop by applying Ohm's law of resistance. Current sensing could also be done by using the flux field generated by the current (Hall Effect Sensor) or by measuring the induced AC current with a transformer. (Giovino, 2010)

A current shunt is a device that allows current to flow through a specific point in the circuit, giving it the ability to alter the amount of current flowing by creating a low resistance path. It works as a proportional measure of current flow by taking the voltage drop across the resistor. This current sensing solution is the most common approach and most feasible. Prior to selecting the shunt, it must be determined whether the current sensing will occur at the supply rail of the load (high side/forward current path) or the ground point for the load (low side/ return current path), as well as the common-mode range, offset voltage, and common-mode rejection ratio. (Giovino, 2010) High-side current sensing is when the resistor is placed in the current conducting path between the power source and the load. This configuration allows for monitoring short circuit and open circuit problems because it is the most responsive to the changes in current flow through the circuit, without creating a disturbance to the ground of the system. However, because the resistor is not connected to the ground, a differential voltage must be measured by precisely adding the right differential amplifier. Low-side sensing is measured by adding one end to the system ground and the other to the load ground. Unfortunately, this simple placement results in a floating ground higher than the system ground, the system is susceptible to noise interference and is unable to detect open and short circuit conditions. (Ziegler, Woodward, Iu, & Borle, 2009)

Selecting the precision, value, and size of the resistor is all dependent on the current we expect to measure. For a current draw of 1 amp or more, larger resistor values are suggested. Although the larger the resistor, the greater the power loss, it will also produce more accurate measurements. To maximize accurate measurement, four-terminal current sensing resistors should be implemented. Two of the terminal ends are for the current path of the application circuit and the other pair are for voltage detection of the sense amplifier. This setup, Kelvin Sensing, ensures that voltage measured is the actual voltage across the resistor and neglects resistance of a combined connection and noise. Low stability over operating temperature is another important factor to consider. (Giovino, 2010)

Dealing with low resistance values can cause issues like thermal EMF. Thermal EMF is a small voltage generated by some difference in temperature across an inductor causing inaccurate readings. Most current sensors can only operate within a certain frequency (100 kHz), because if it is too high, too much current will be induced, and the system will overheat. These sensors also have an inherent voltage offset and thermal drift, so they require support from integrated ICs. (Giovino, 2010)

Mechanical Requirements

Repeatability and consistency are fundamental to the modular robotic arm concept. Therefore, the compact mechanical system in each link must be designed to minimize electromechanical error of any kind. The consistency between the physical parameters and the values that sensors report is critical to ensure that the system follows the kinematic models accurately. The forms of mechanical inaccuracy must be explored.

Joint Torque Relationship

$$\tau = R \times F$$

Torque is a function of force multiplied by the distance from the pivot of rotation. Due to the modular nature of the robotic arm, the mass calculations change depending upon the physical configuration of the system. To ensure that the joint of the first link can handle the external moment in any possible configuration of the system, the maximum possible moment of the base joint must be determined. The greatest moment about the base joint would be produced in a configuration in which the end effector is at the edge of the task space at the furthest distance away from the base joint. The output torque of the link’s joint must be well above the maximum moment exerted on the joint. This relationship is vital to the success of the mechanical design. Fundamentally, motor selection must be determined while considering the ratio of the mass of the motor to the output motor torque. Since the mass of the motors is a large portion of the total mass of each joint, using a lighter motor will exert a smaller moment on the joint.

System Backlash

To produce a physical system that mimics the kinematic models best, all sources of mechanical error must be minimized. In a mechanical system with a gearbox, the output is bound to have backlash in some capacity. Backlash is the amount that a gear tooth space exceeds the width of a gear tooth engaged in mesh. (Niijjaawan & Niijjaawan, 2010) Backlash is problematic because it is mechanical leniency in the system that significantly reduces precision of control. There are multiple types of backlash, as follows. The gearbox must be designed to minimize all forms of backlash.

- Circumferential Backlash: the arc length that the gear is rotated until the meshed tooth makes contact while the mating gear is still.
- Normal Backlash: the minimum distance between each tooth in a pair of meshed gears.
- Angular Backlash: the maximum angle that the gear can rotate while the mating gear is still.
- Radial Backlash: the displacement in the center distance between a pair of meshed gears

Therefore, exploring methods to reduce mechanical backlash is essential to the success of the electromechanical design. Methods and design considerations by which backlash is minimized are discussed in detail in the design instantiations.

Automatic Joint Locking

The joint gearbox must accommodate mechanical fault tolerance in the event that a motor fails. As discussed prior, this design must achieve mechanical fault tolerance by locking the joint in place when the system realizes a motor has failed. Automatic joint locking is imperative to holding all subsequent links in their desired location when a motor fails, otherwise a single motor failure could cause the entire system to lose its mechanical rigidity. Automatic joint locking can be accomplished through either a passive or an active locking mechanism. An active locking mechanism would require an additional exterior locking mechanism to communicate with the motor feedback signals. A passive locking mechanism does not require any feedback from the system, and it would automatically lock the joint without an additional mechanism. The active locking system will add more complexity to the overall design. To ensure that the locking mechanism is instantaneous, it is best to use a passive locking mechanism.

Passive Joint Locking

A worm gear and worm wheel pair can be used to prevent back driving and achieve passive self-locking. This solution appears to be the most compact, least expensive, and easiest to troubleshoot and maintain. However, worm gear and worm wheels prove to be very expensive compared to traditional spur gears. Therefore, key qualitative and quantitative specifications must be considered in the worm gear pair selection process. In order to be passively self-locking, the following must be true:

$$f < \cos(a)\tan(b)$$

f = coefficient of friction between worm gear and worm wheel

a = pressure angle of the worm wheel (driven gear)

b = lead angle of the worm gear (driving gear)

Typically, if the lead angle of the worm gear, the driving gear, is less than 4 degrees, the worm gear pair cannot be back driven. (Zhan, 2016) Yet, we must contact original equipment manufacturers (OEMs) to verify such performance data.

Similar Projects

TRACLabs

TRACLabs, an exploratory research lab out of Webster, Texas, has over 20 years of experience in robotics and automation. In 2011, they developed a seven-degree-of-freedom reconfigurable manipulator (RMM) to be used on analog rovers for NASA planetary exploration based on the SBIR program. This robot comes with very special features like identical Universal Mating Adapter (UMA) coupling, which allows for secure data communications across the interface, power transfer, and mechanical attachments between all joints from base to end-effector. Other features include internal power and data lines, individual control per joint, failsafe magnetic break, non-volatile memory that stores relevant parameters, and simple detachment of interchangeable links. (Burridge, 2011)

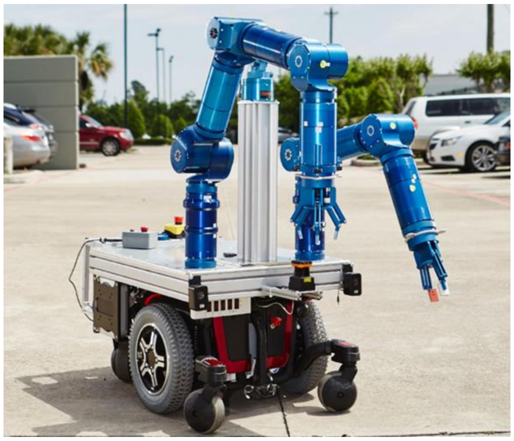


Figure 1 – Reconfigurable Modular Manipulator from TRAClabs

MIRA MQP

The Modular Interchangeable Robotic Arm (MIRA) project of 2018 is the MQP that is most similar to our project. The project modified the existing RBE3001 robotic arm to use a modular control system, added a removable fourth joint, and created a real-time visualization GUI. The goal of this project was to “make robotics education accessible to a larger number of people, without compromising the potential of each person to gain a high quality understanding of the way robotic arms behave” (O'Shea, Taglieri, & Titus, 2018). This project implemented a lot of relevant features, but with a different goal in mind. Key takeaways include:

- 3D printed prototypes are critical for rapid mechanical development
- Hall effect sensors are an effective way to measure motor current
    - A ring buffer with an exponential filter solves noisy signals
- Having many level shifts between 3.3V and 5V unnecessarily increases circuit complexity

Other MQPs

Other MQP projects were found to be informative for the development of this project. The Blisk Inspection System project of 2018 used a prebuilt ABB robotic arm to inspect blisk root fillets for GE Aviation. The Piano Playing Robotic Arm of 2019 used a prebuilt industrial arm and a custom hand-shaped end effector to play a piano expressively. The Real-time Control of Robotic Arm Based on Hand Tracking Using Leap Motion Sensor Technology project of 2015 took input data from a LEAP Motion system about how a human hand moves, then translated that to a pre-built 6-DOF robotic arm in real time. All these projects used pre-built arms, but their approach to controls and software is still relevant. Key takeaways include:

- Ensure the communication protocol is consistent on both ends, at both the high and low level
- Include the necessary feedback sensors for a proper control loop
- Resolve external dependencies as fast as possible

Task Specifications

- Manipulate 1 kg payload
- Operate within a hemisphere with radius of 2ft
- Navigate to position within 5% when not in a failure state
- Navigate to position within 7.5% when in a failure state
- Adapt to module failure during operation
- Mechanical support for up to 6 link, plus base link and end effector
- Software support for up to 30 links
- Communication protocol supports status updates at 100 / second
- Communication protocol supports initialization within .25 seconds
- Communication protocol fails to meet deadline no more than .001%
- Put safety in place to prevent short circuiting
- Ensure secure wire connections
- Operate with power dips of no more than 27%
- House 90% of the electronics

Methodology

Communication Architecture

Communication Protocol

When deciding how our links would talk to each other, we initially considered SPI and I2C. However, it quickly became apparent that these protocols were not ideal for our application. The problem with SPI is that it requires an additional select line for each node in the system. This would result in the number of links being capped by how many select lines the base computer can fit. And this restricts messages to be from only the base computer to each node, not between the nodes. Inter-node communication would require an exponentially increasing number of lines with each additional link. I2C does not require additional lines for each node in the system because it uses an addressing scheme. However, devices that support I2C have dedicated hardware for it and use a hardware-defined address. This is a problem because our addressing scheme should tell us where in the chain each link is. If our addresses are hardware-defined, then if we swap two links our system won't be able to tell the difference. Our addresses need to be dynamic and software defined.

The next idea we considered was using a simple serial UART to talk between nodes and have the nodes forward messages between each other. Essentially, creating a small packet-based network that is loosely based on the ethernet protocol. This way, we can have software-defined addresses that are assigned via a basic form of DHCP where each link gets the address of the previous link plus one, and we get the throughput benefit of a multi-hop network. However, this presents several serious challenges, such as how to handle traffic congestion since most packets would eventually be sent over the link between the first node and the base computer. These problems are all well-defined and solved by the networking protocols we use every day such as TCP, however implementing TCP is a non-trivial task and would be overkill for this project.

We then explored using a CAN bus, which is a communication protocol that uses broadcast messages with priority message IDs. Since it is a bus and not a multi-hop network, there is no issue of traffic congestion other than at the end points, but the CAN protocol includes features to slow down data transmission in case of congestion issues as well as message acknowledgement, retransmission, and error detection. The CAN protocol solves all the issues present in the UART packet network idea; however, it introduces a couple of problems of its own. Since each message is broadcast and has a message ID, we need a way to target messages to each node. Because the message ID is a priority, if we use the target node ID as the message ID, then any messages going to the early nodes will receive priority over any other messages, which is an unfair system. Additionally, each node will read every message and must determine if it is important or not in software. These problems can be solved by using the CAN protocol as a layer 2 network and implementing a layer 3 protocol top of it. This is the same as defining IP on top of Ethernet. We get the benefits of the UART packet network, but without any of the major challenges.

The CAN protocol uses an 11-bit header and supports up to 8 bytes of data per message. If two nodes attempt to transmit at the same time, the lowest header number will win. Because of this arbitration process, no two nodes should attempt to transmit data with the same header number. It is natural, then, to have the ID of the sending node be embedded in the 11-bit header to ensure that no two nodes have the same header. A 5-bit node ID allows for 30 unique nodes in the system, with *00000* reserved for the base computer and *11111* reserved as a broadcast address. The 5-bit sender ID will be the least significant bits of the header, with the most significant 6 bits being a priority. This way, higher priority messages (with a lower number) will always override lower priority messages, and the sender ID is just to give an order to prevent header collisions during arbitration. In the 8 bytes of data per message, the first byte will be used as a header byte. The first 5 bits will be the receiving node's ID, the next 2 bits will be a data ID, and the final bit will the final fragment flag. The data ID and final fragment flag combined tell us which CAN messages combine into one data packet for the application. When we receive a CAN message with the final fragment flag, we combine the data from the previous messages with the same data ID into the data packet for the application. If we receive a new CAN message with a different data ID and no final fragment flag, we ask for a resend.

The system is designed to be able to have every node sending updates 100 times per second. In the most extreme case, this is across all 30 nodes. With the CAN header taking 51 bits and the data header taking 8 bits per message, that is 59 header bits per CAN message. The most amount of data bits in a message is 56, which results in a data to header ratio of 48.7%. Figure 2 shows the Data Efficiency of the packet by the number of bytes to be sent. It caps at the maximum every time all 7 bytes are full in the final CAN message of the sequence.

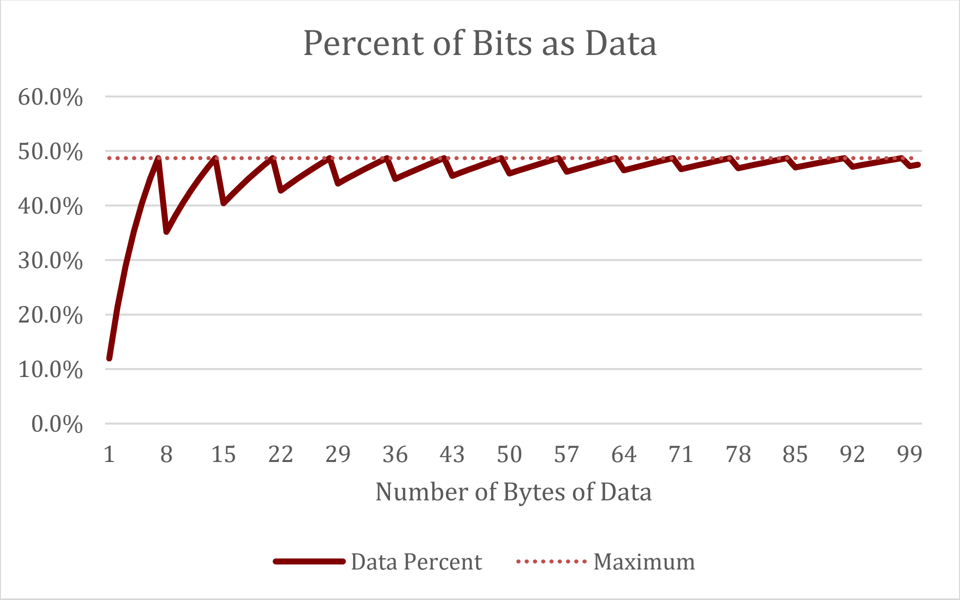


Figure 2 – Data Bit Efficiency by Packet Size

Because a status update from a node is not a lot of data, it can be sent in 7 bytes or less. The only information needing to be constantly sent is the angle of the arm and the data from any sensors added to it. With the current design of no sensors other than the encoder, this can be accomplished in 2 bytes, the first for a version number of the packet format and the second for the actual encoder value. To keep this extensible, 7 bytes of data will be considered for each message in case other sensors are added and this is one of the maximally efficient amounts of data. The CAN protocol can support up to 1MB/s of data transfer. Divided by the total message size of 115 (59 header plus 56 data), this is 8695 total messages per second. From each of the 30 nodes, that is 289 messages per second. However, this is only half of the messages. The base computer also needs to send updates to each node at 100 times per second. Each node only requires the desired angle, but to keep the math simple and overestimate for safety, 7 bytes of data will again be considered. For both sending and receiving updates, each node will be transmitting 144 messages per second at maximum capacity, which is well above the desired 100 updates per second.

Microprocessor Selection

We currently have multiple MSP430F5529s from previous classes, and the initial thought was to use them. However, they are too large of boards to fit in our links, so we had to find a smaller board. The primary requirements were size and number of pins. The board had to be able to be programmed over USB or FTDI for rapid development, and required a SPI interface for the CAN module, as well as a UART interface for debugging and more digital IO pins for controlling the motor and reading sensors. Arduino boards are designed with ease of development and space as design considerations. The Adafruit Trinket is one of the smallest boards available, however it did not have enough IO pins and was not fast enough. The Trinket Pro had more IO pins and was faster, but it had been deprecated in favor of other boards. We decided on the ItsyBitsy 32u4 because of its high number of IO pins and small size, while using a processor of similar power to that of the larger Arduino Uno. Another benefit of using an Arduino-compatible board over the MSP430 is that there exists a software library for interfacing Arduinos to the MCP2515 CAN module. There does exist a software library for interfacing the MSP430 to the TCAN4550 CAN module, however it was written for a slightly different MSP430 than the one we had, so it would have required some tweaking for it to work properly. The development for the Arduino board was likely quicker because of this.

Kinematics

Algorithm Design

The first instantiation of the kinematics algorithm as applied to the robot uses a combination of the Damped Least Squares iterative solving technique, and trajectory generation. The resulting algorithm generates joint space targets that follow a trajectory from the current arm position to a specified task space coordinate. The algorithm is scalable to any number of links, and accounts for damaged or otherwise non-functional links by disabling their movement and solving as if they were a static link. With the use of worm gears to prevent back driving of motors, non-articulated joints can be expected to remain static. The damping coefficient used in DLS is scaled by the minimum singular value. This allows the damping to be very low at configurations that are not close to singularities but be increased to compensate for errors as the arm nears a singular configuration (Di Vito, Natale, & Antonelli, 2017).

The damping coefficient λ₀ can be chosen at runtime to configure how damped the algorithm becomes.

$$\lambda = \begin{cases} .1 & if\ \sigma_{min} > 1 \\ \lambda_0/\sigma_{min} & if\ \sigma_{min} < 1 \end{cases}$$

The resulting dynamic damping coefficient λ also provides robustness to the algorithm when operating with a variable link count. The arm solving process current starts with the arm in a position where all joints are turned slightly, as the arm faces a slowdown when trying to calculate points from a straight configuration. Once in the trajectory the algorithm solves from the previous location to the next location, as the proximity reduces the number of iterations needed to converge greatly.

The algorithm also runs a real time simulation, visualizing the arm in a stick model fashion as it converges on setpoints. The simulation can be configured to solve a sequence of setpoints for demonstration or receive inputs in real time for operation. Before being sent to the solving algorithm, these set points are passed through a trajectory generation function. Using a quintic polynomial, the setpoints are interpolated in task space form the arms current position to the desired end position. The trajectory is configured so that both velocity and acceleration start and end at zero. The resulting trajectory has a smooth start and end of end effector motion, helping the arm to avoid overshooting the target position. An example of how a joint space trajectory effects motion can be seen in Figure 3. Note however, that our algorithm creates task space trajectories, interpolating over the XYZ coordinates of the end effector locations rather than the joint angles.

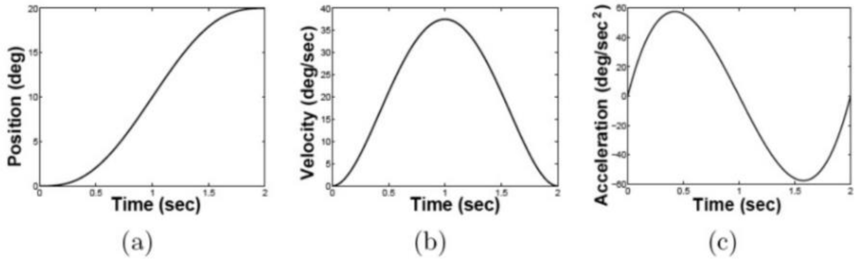


Figure 3 – Example trajectory curves

The algorithm also accounts for the joint angle limitations of each joint. A joint exceeding its limit within the algorithm simulation will prevent the result from being sent to the arm, and the algorithm will continue to run with an added bias on the over-rotated joint that pushes the angle back into acceptable values. Using the algorithm and evaluating the time taken to reach points, several points were tested in order to establish a minimum distance from the arm base for the task space. The operational area begins at approximate one-third of a joint piece's total length. The outer extent of the arms workspace will be determined by torque limits.

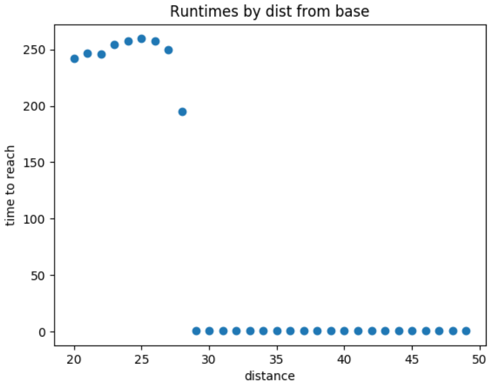


Figure 4 – A sharp spike in runtime indicated the minimum bounds of the operational area

Due to the minimum distance from base and joint angle limitations, the arm does not typically bend enough to collide with itself. To detect the fringe cases where this does occur, the arm measures distances between links and flags them when they encroach upon each other. Flagged solutions cannot be used as the arm would collide with itself, and the algorithm is run form a new random start to achieve a slightly different solution.

The kinematics system is also capable of calculating torques on each joint. Weight of the arm itself is precalculated and known end effector weights in the event of grabbing objects can be added in. Calculating and maintaining information on the arms weight and torque distribution can be used to bias the kinematics towards more efficient weight distribution. For example, bending a joint close to the base more, and a similarly oriented link farther out less, the center of mass of the arm would be closer to the base, easing the torque on relevant joints. Joint torques are also used in the arms control loop. Joints experiencing high downward torque from the weight must be sent a stronger signal to balance out the downward force. By implementing a torque-based control loop, the arm can be set to properly compensate for weight by resisting the downward pull with a relatively equal upward force form the motor. By combining this control loop with a standard position-based control, the joint can navigate to points in an upwards and downwards directions with similar stability.

$$u(t) = Pe(t) + I\int_0^t e(t)dt + D\frac{d}{dt}e(t) - G\tau$$

This equation represents the control equation used by the simulation. P, I, and D are the PID values and G is the coefficient for gravity compensation.

In wake of the COVID19 pandemic, the simulation was broadened to also include the control of the robot motors to account for the lack of a physical robot. Motor signals are generated by a control loop that integrates gravity compensation into a standard PID control system. In between runs of the kinematics algorithm, the arm simulation will now perform smoothed motion to set points.

Link Parameters

Each link reports two transformation matrices. One from the end of its arm to the joint, and another from the joint to the end of its other arm. We can use these matrices to calculate the transformation matrix between any two joints as well as the full transformation matrix to the end effector.

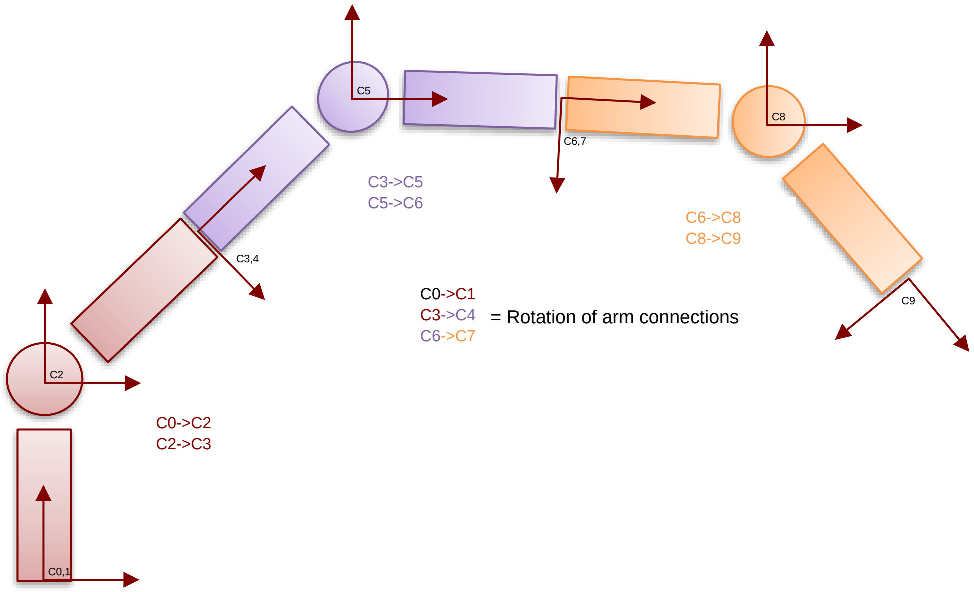


Figure 5 – Example Coordinate Systems of a Three-Link Arm

Shown in red in Figure 5 are the different coordinates to consider for this chain of 3 joints. The reported transforms are shown in the corresponding colors. Each link has 3 total coordinates it is considering. As an example, let's look at the middle link. C_3 is the coordinate of the end of the previous link and C_4 is the coordinate of the start of the middle link. These will be the same in 1 of the 4 configurations, but since our arms can attach at different 90° rotations, we will need a transform from C_3 to C_4 to express that rotation. We then have a transform from C_4 to C_5, which is at the joint. The link will report the overall transform from C_3 to C_5. The last coordinate system associated with the middle link is C_6, and the link will report the transform from C_5 to C_6. The main computer can then construct the transformation matrix up to any joint, or the end of the arm by just multiplying the transformation matrices given by each link in the arm.

Electronics Component Selection

Shunt Resistor

Since this project does not have demanding performance requirements, Ohm's law of resistance technique was implemented for its reliability, feasibility, and simplicity. The shunt connects between the supply rail and the load, measuring the forward current path. This high-side shunt positioning is the most responsive to the changes in current flow through the circuit, allowing open and short circuit conditions to be monitored. The current sensor that was selected to handle these requirements is the INA260 High or Low Side Voltage, Current, Power Sensor from Adafruit. This breakout board measures high or low side DC current, the bus voltage, and automatically calculate the power. It works with microcontrollers, compatible with 5V logic, measures voltages up to 36 VDC, and currents up to 15 A continuous. This breakout board was the best option because it does not have the constraint of low side measurements and can do the work of two multimeters. Since a mechanical modification forces the joints to lock when there is a large electrical malfunction, preventing a current spike, the sensor does not need to be implemented on each link. One sensor was placed at the base so that the current conditions throughout the whole arm can be monitored to prevent approaching open and closed-circuit problems.

Figure 6 – INA260 Current Sensor

Motor Driver

Although the motors have encoders, allowing them to control motion parameters, an H bridge motor driver is necessary for the motor to turn forwards and backwards. An H bridge is an electronic circuit switching polarities of the voltage that is supplied to the load. The H bridge must be able to handle up to roughly 7 A, which is the optimal performance for each motor. It was preferred that the input voltage support a range from 5V (the power supplied to electronics in the system) or 24V (the power supplied to the motor). Size is an important factor to consider. Since a driver is necessary for each link, it cannot be too heavy nor too large, and would need to fit inside of the link with other electronics. The MD10C Rev2.0 motor driver by Cytron Technologies, used to drive high current brushed DC motors, was initially selected but had to be replaced due to size constraints by the MD114 MOSFET Trigger Switch Drive Module from Anmbest. This driver is smaller and more powerful with the ability to handle 15 A continuous, enhanced cooling conditions, and parallel active output.

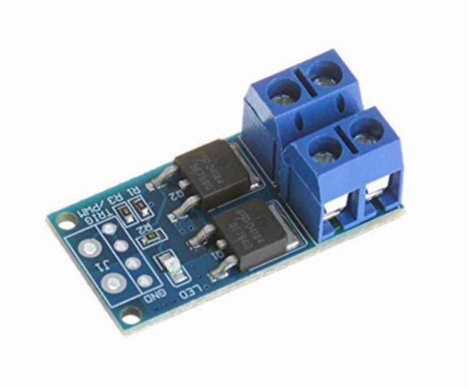


Figure 7 – MC114 Motor Driver

Power Supply

After deciding the voltage and current demand for the larger components of the arm, the next step was to figure out the best way to power up the system. With a requirement of 24 volts per motor and 13 amps stall current per motor, it was necessary to find a large power supply that could handle high current. Initially, research was done to connect the arm to a 120 VAC wall outlet. Between the wall outlet power cord and the arm would be a step-down converter, but the converters that were necessary to run the amount of current necessary were far too expensive and too large for the project. It was decided that a bench power supply, that is readily available in the labs, would be strong enough to power to handle the arm at full functionality. The GPS-3303 Bench Power Supply by GW INSTEK is a 195W linear DC Power Supply with overload and reverse polarity protection, and a high regulation (0.01% + 3mV) and low ripple/noise (< 1mVrms, 5Hz to 1MHz) maintained in constant voltage mode. The GPS-3303 has three output terminals per device, each containing 3 amps, which is a total of 9 amps per power supply. If one supply is not enough, then two can be connected to produce a current of 18 amps, and so forth. Power supply systems with enough current to support this project are quite expensive, ranging from $1000 – $1700. This choice was the most beneficial over the AC/DC converter and purchasing a new bench power supply because of its easy access when needed for testing and free availability. When selecting the gauge of the wire desired to use that will run power through the system, it is important to consider the max current draw that the robot will pull. With the four motors drawing the most current and preforming at optimal current (6.5 A per link), it was decided that a 12 AWG wire would work well to manage the current demand of the robot and carry the 24 V throughout the system. At the ends of each link, there are four Molex 2-Circuit connectors (one side with all female connectors and the other side with all male connectors) to run the power ground and data lines. Although the main voltage supply is 24 VDC, the motor driver, CAN Bus communication, and microprocessor require a 3.3V – 5V input. A linear voltage regulator or buck converter can be used to decrease voltage from 24 V to 5 V. Although a voltage regulator would be a considerable option, the amount of heat dissipated within an enclosed space could damage other electronics, requiring a large heat sink, whereas a buck converter can regulate its temperature and is very power efficient. The LT 1766EFE is a

200kHz giant buck switching regulator that can handle an input voltage of up to 60 VDC, high efficiency at 1.5 A, and includes an oscillator, control, and logic circuitry.

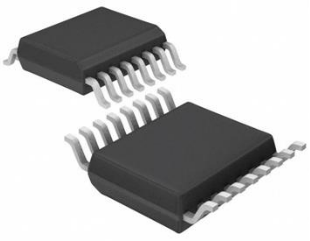


Figure 9 - Buck Converter

Figure 9 – GPS-3303 Bench Power Supply

Mechanical Design

Motor Selection

The motor for the joint gearbox must be capable of overcoming the moment of the attached subsequent links. This is precisely why the mass of the motor plays a key role in the dynamics of the overall system. The weight of the motor makes the motor selection process tricky because stronger and higher voltage motors also cost much more. In the selection of this motor, the outliers with higher output torque respective to their mass are most desirable. Selecting a motor based on this output torque to motor mass ratio was imperative.

Initially, our team contacted Harmonic Drive to discuss a possible academic partnership. Despite multiple weeks spent communicating with their sales team, the prices of their high-performance motors vastly exceeded the limits of our MQP budget. To overcome difficulties in finding an affordable lightweight high-torque motor, we decided to minimize mass in other aspects of the design. The selected motor is a Devantech EMG49 24V gear motor 49:1 with an integrated encoder. This motor appeals to our design considerations because of its high torque to low mass ratio. This selection is cost effective at about $77 per unit.

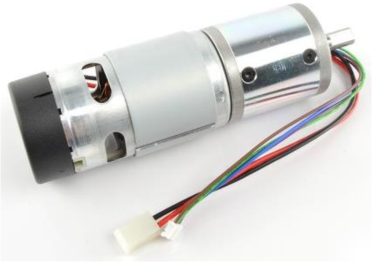


Figure 10 – Devantech 24V, 49:1 Gear Motor w/ Integrated Hall Effect Encoder

The motor performance curves are show in Figure 11. The motor specifications are as follows:

	No Load State	Rated State	Stall State
RPM	143	122	0
Current (A)	0.5	7	13
Torque (oz-in)		222	1472
Torque (N-m)		1.57	10.39

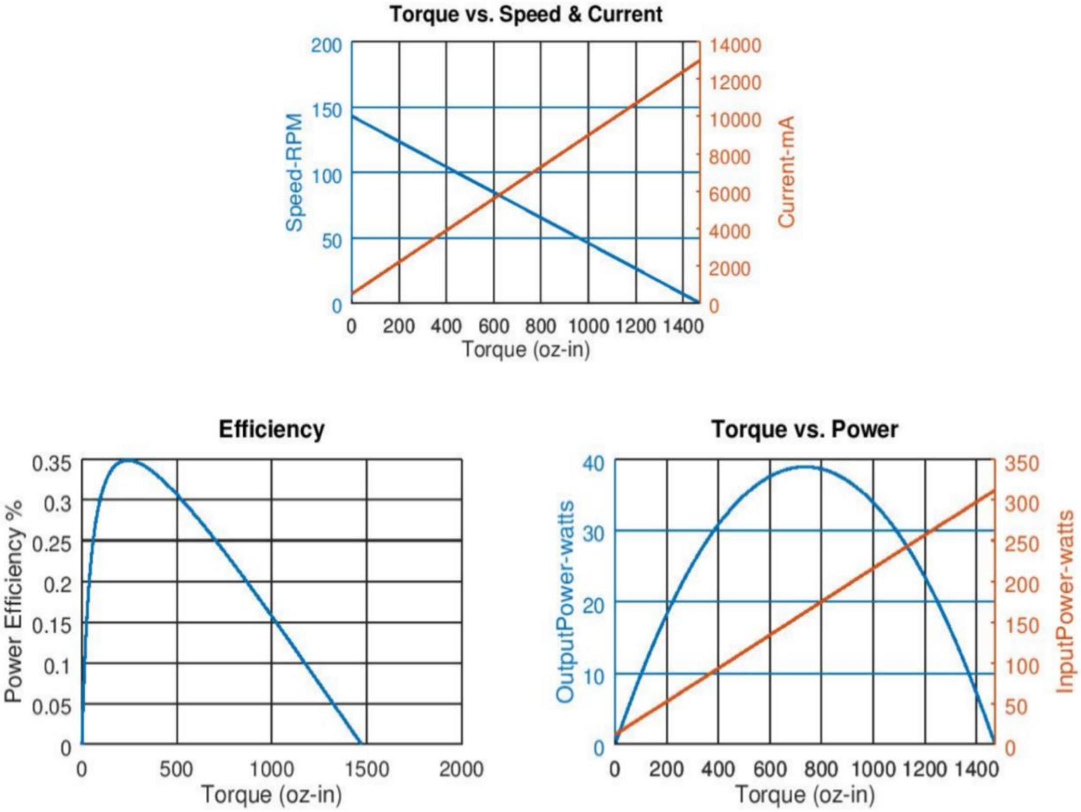


Figure 11 - Devantech Motor Performance Curves

Worm Gear Threading Selection

The benefits and drawbacks of implementing a single vs. double threaded worm gear must be explored. Frankly, our team was not initially aware that the worm gear threading could be modified to change the performance of the worm gear pair. During this selection process, enveloping worm wheels were considered as well.

Double threaded worm gears have threads twice as tight to increase efficiency up to 70% and slightly reduce backlash. (Kubota & Nyumo, 1963) However, the tighter threads cause the nominal lead angle to be higher than 4 degrees. The lead angle increases with the number of thread starts on the worm gear, as shown in Figure 12. Any worm gear with a lead angle higher than 4 degrees can be back driven. Thus, double threaded worm gears cannot be suitable for our self-locking gearbox application.

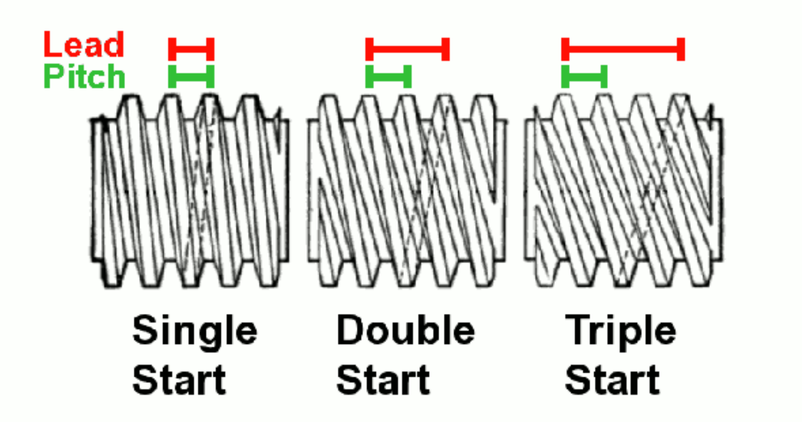


Figure 12 – Single thread vs multiple thread worm gears

An enveloping worm wheel is a shaped worm wheel designed to have more surface area contact with the worm gear. Enveloping worm wheel have a curved throat to mesh the gears closer together, as shown in Figure 13. Enveloping worm gear pairs have among the lowest backlash of all kinds of gears. Naturally, enveloping worm wheels are the most expensive kind of worm gear to manufacture. This level of precision is unfortunately far beyond the budget of this Major Qualifying Project. Nevertheless, it is necessary to explore all possible design considerations to minimize backlash for this application.

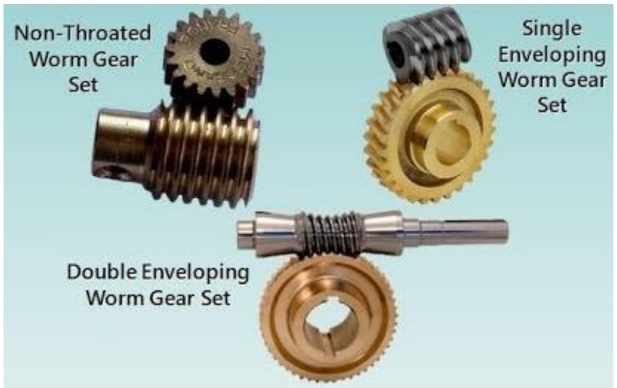

Non-Throated Worm Gear Set
Single Enveloping Worm Gear Set
Double Enveloping Worm Gear Set

Figure 13 – Non-enveloping worm gears vs. enveloping worm gears

Hence, after considering particular gear specifications to mitigate backlash and achieve back driving, the single thread worm gear and worm wheel pair seems to be the best fit for our task specifications in this design. In order to properly accommodate the worm gear and worm wheel in the joint gearbox housing, the locations of the thrust bearings must be identified first.

Three Motor Gearbox – Design Instantiation

A potential design for the fault-tolerant gearbox is modeled in Figure 14. A worm wheel coupled with a worm gear is implemented to provide passive static locking at the joint. Zero back driving by static locking in a worm gear and wheel pair is only possible if the friction angle is greater than the lead angle of the worm gear. In an instance in which a motor fails or loses power, the joint will lock its position while the external shaft encoder reports its position. Figures 15 and 16 display the front and right views, respectively. However, this three-motor design instantiation depicted was never adopted. The three-motor gearbox concept also added three times higher likelihood of mechanical failure. Due to the planetary gearing system if one motor failed and locked in place, the other motors would need to overcome the torque requirement of back driving the failed motor. Instead, a lighter and more compact single motor gearbox design was implemented.

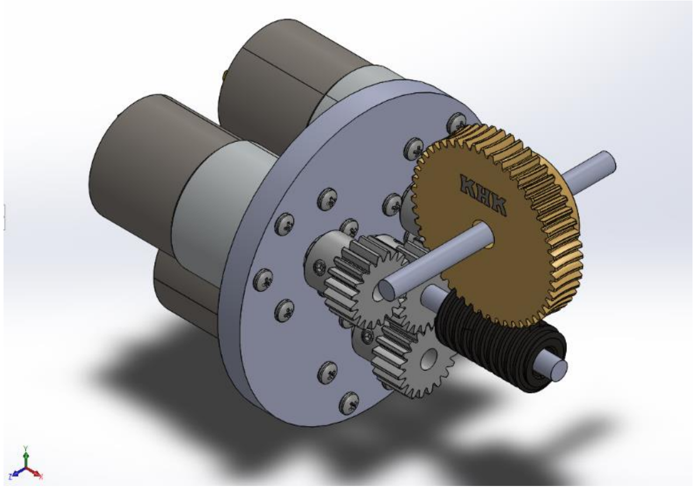


Figure 14 – Modeled Fault-tolerant Gearbox of Joint (3 motor instantiation), Isometric View

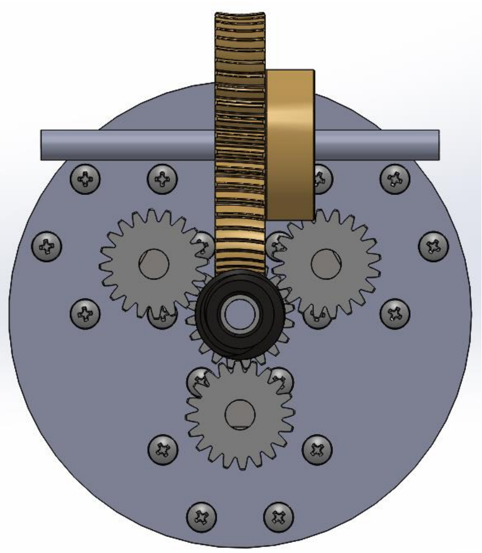


Figure 15 – Modeled Fault-tolerant Gearbox of Joint (3 motor instantiation), Front View

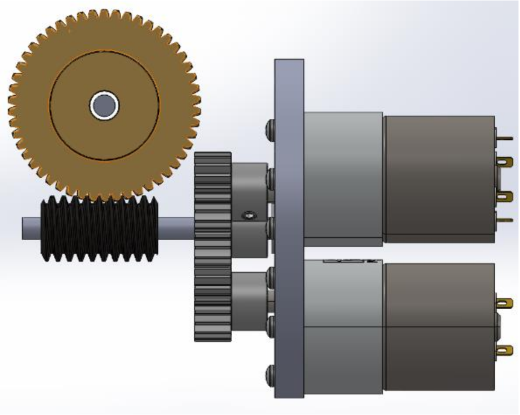


Figure 16 – Modeled Fault-tolerant Gearbox of Joint (3 motor instantiation), Right View

Single Motor Gearbox – Design Instantiation

A superior design instantiation accommodates the desired task specifications with a single Devantech 24V motor. This aspect of the design is superior because it significantly reduces the likelihood of a motor failure threefold. The motor shell houses the entire motor with additional room in the back for the motor driver and other electronics. As shown in the exposed SolidWorks prototype figures, the gearbox is quite compact and housed with thick ABS plastic. The prints will be completed with 100 percent infill to ensure a tight fit around the bearings and provide ample support to the overall structure. The worm wheel, KHK BG1.5-20R1J8, and worm gear, KHK SW1.5-R1J8, are mated orthogonally as shown in Figure 17.

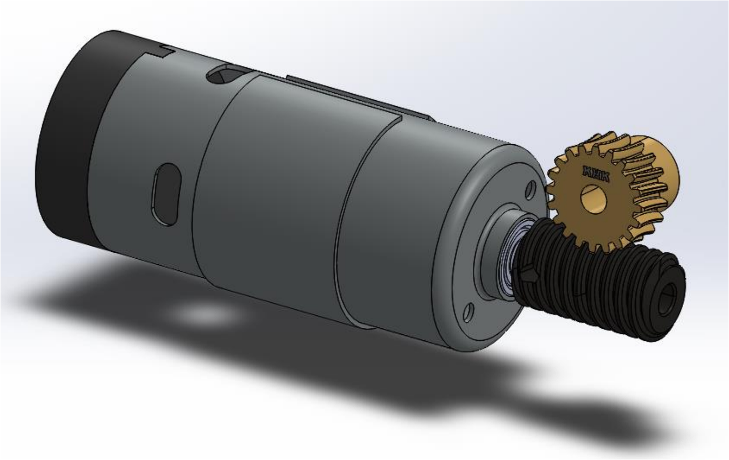


Figure 17 – Modeled Mate between Worm Gear and Worm Wheel (1 motor instantiation)

As shown in Figure 18, the thrust bearing has yet to be included in this model. The thrust bearing is required to handle the lateral forces the worm gear will experience.

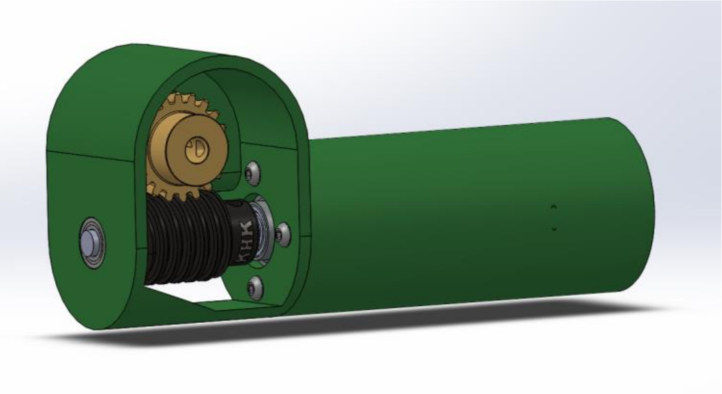


Figure 18 – Modeled Exposed Gearbox with Motor Shell (1 motor instantiation)

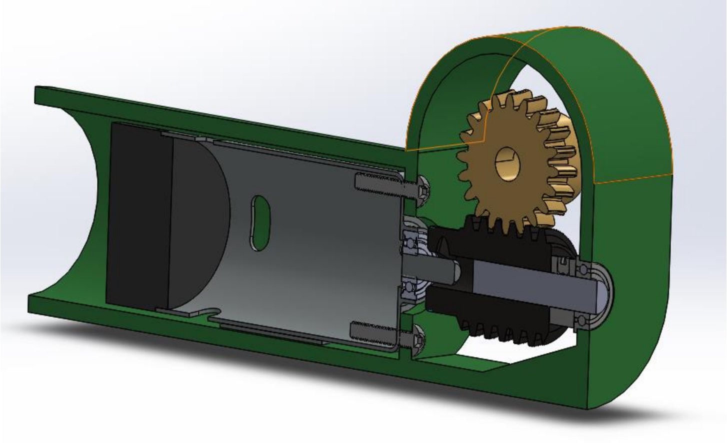


Figure 19 – Modeled Gearbox with Bearings, Midsection View (1 motor instantiation)

The section views in Figures 19 and 20 show the short axle, 8mm Stainless Steel Rotary Shaft 1265k64, that extends the motor shaft so that it can be supported by the bearing, Flanged Ball Bearing 7804K147, opposite the motor. This is imperative to provide support along the entire span of the worm wheel. The thicker plastic shown on the left side of the worm gear in Figure 18 is intended to provide additional support for the rotational force of the joint. The thrust bearing, McMaster Carr 2011N112, is shown in grey resting against the vertical support wall in Figure 20. To reduce friction in the gearbox system, high-viscosity lithium grease will be applied directly to the worm gear and worm wheel. Minimal grease will be necessary on the shafts themselves since rotational friction is already significantly reduced by the bearings.

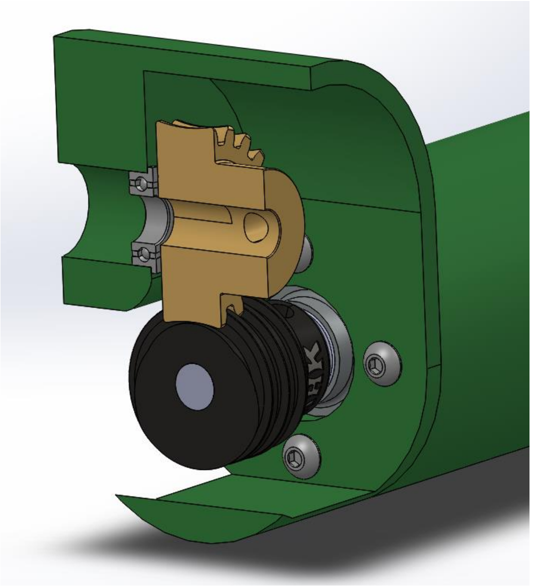


Figure 20 – Modeled Gearbox, Front Section View (1 motor instantiation)

Link End and Electronics Housing

Once it was decided that the team would proceed with the single motor design instantiation, the rest of the requirements of the 3D printed housing were developed around it. This design process began with completing the gearbox design by coupling the shaft with the other end of the link. The link end is much simpler than the gearbox end of the link as it serves merely as a connection to the following link and as a housing for the electronics. The electronics are housed in the link end include the ItsyBitsy microprocessor, CAN module, and shunt resistor. An isometric view of the link end (yellow) coupled with the gearbox end (green) is shown below in Figure 21. The vertically oriented M4 screws are screwed into the shaft collar, Set Screw Collar 57485K67, of the link end to rigidly mate the shaft to the ABS plastic housing. The exposed interior shown in Figure 22 allows plenty of space for the electronics to be housed compactly and protected from entanglement.

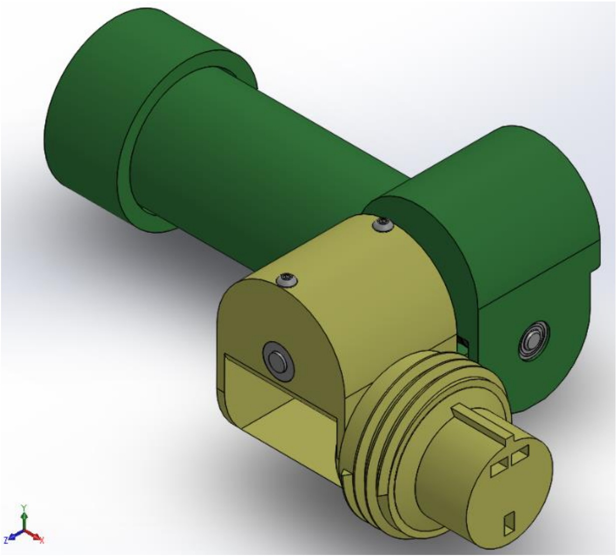


Figure 21 – Modeled Link End (yellow) with Gearbox End (green), Isometric View

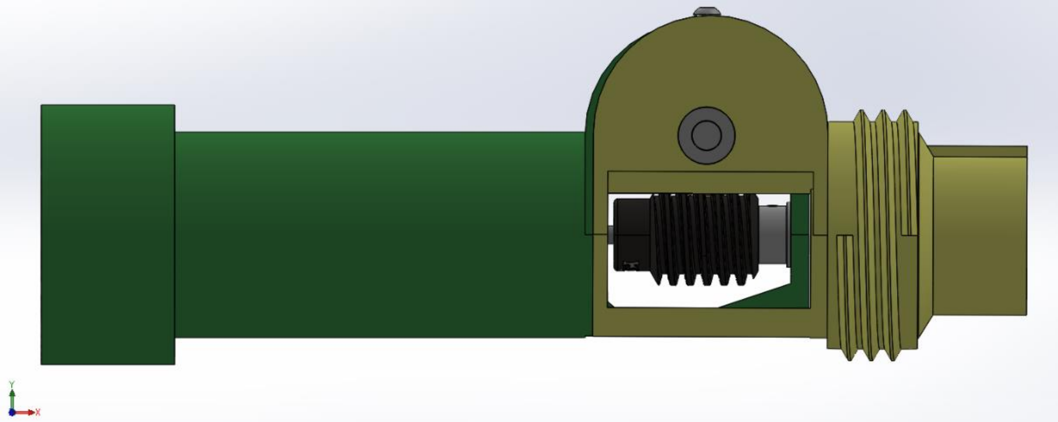


Figure 22 – Modeled Link, Side View

Link-to-Link Rigid Electrical Connection

To achieve the project objective of true modularity, each link must be capable of connecting to another link in more than one orientation. Each link has a range of motion in one degree of freedom. Thus, the links must be able to attach to each other directly as well as an angular offset reach in more than two dimensions in the robot task space. In order to reach this goal a dummy-proof custom port was created so that there would be no risk of an electrical short circuit when connecting links at a 90-degree offset.

There are two key aspects to the link-to-link rigid connection: the interface between the links end-to-end and the threaded nut that holds the two neighboring links together. The custom physical port was modeled with a single male notch and two female notches to allow two configurations in which the links could be connected. The female port with two sets of electrical ports and two female notches is shown in Figure 23. The male port with one set of electrical ports and one male notch is shown in Figure 24. The accident proof concept was implemented by ensuring the physical connection would be robust at an angular offset, but the electronic port would not accept malignant connections. The keyhole-shaped protrusion of the male port (green) is specifically designed to align with either of the two keyhole-shaped slots of the female port (yellow). Since both link ends are circular, the protrusion slides into the slot to protect against external moments around the X-axis of the link. Each set of electrical ports has three two-wire Molex connectors. One Molex is for power, another for ground, and the third for data. The Molex protocol was used to protect against high current surges since 12 American wire gauge is being used. The Molex connectors are not displayed in these models, instead the appropriate dimensions for the Molex fittings are shown.

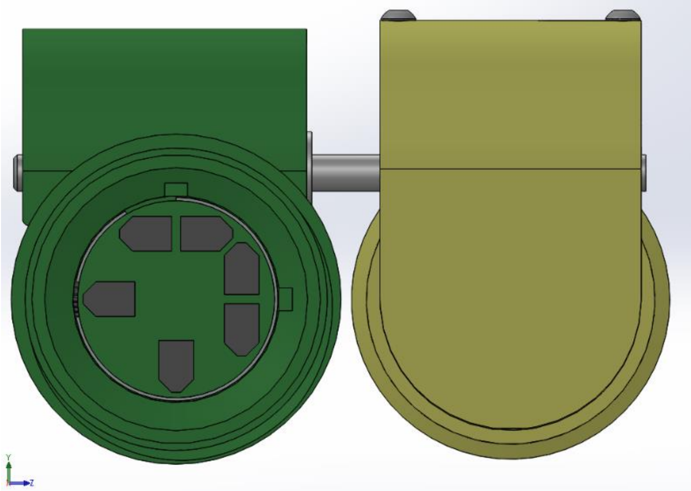


Figure 23 – Modeled Link Female Port (green), Side View

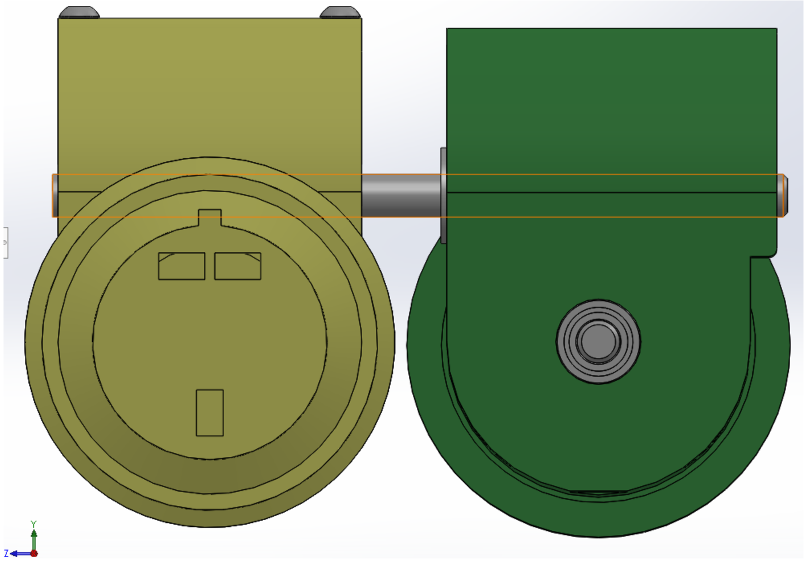


Figure 24 – Modeled Link Male Port (yellow), Side View

The next aspect of the link-to-link connection is the threaded nut that holds the links together laterally. The nut is 3D printed as part of the female port part and is physically constrained from being removed from the female port. When the part is printed as a single body, the dissolvable support material will separate the threaded nut from the rest of the housing. The nut threading pairs with the external threading on the male end of the link. This design mimics that of a hose gasket nut that is rotationally fixed to the end. This threading is best shown by a midsection view as in Figure 25. In this figure, the male end of the link (yellow) is rigidly fixed to the female end (green) by the threaded nut (red) which is printed as part of the female port. The threaded pair was designed with considerable tolerances to allow smooth operation despite minor inconsistencies in the 3D printed material smoothness. A small amount of viscous silicone lubricant would be applied to the internal threading of the nut to reduce rotational friction. The high viscosity lubricant will ensure that the grease will not leave the limits of the threading and contaminate the exterior profile of the link. To hold the links together, the threaded nut design was chosen as opposed to a latch mechanism because it is mechanically simpler and much easier to produce with additive manufacturing methods.

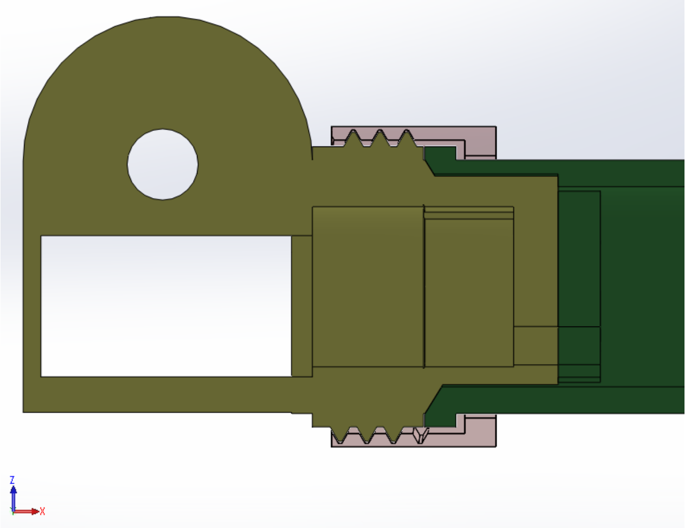


Figure 25 – Modeled Link-to-Link Threading Attachment, Midsection Front View

The interior threading of the ABS plastic nut can also be visualized with the electrical port fittings. This view is shown in Figure 26.

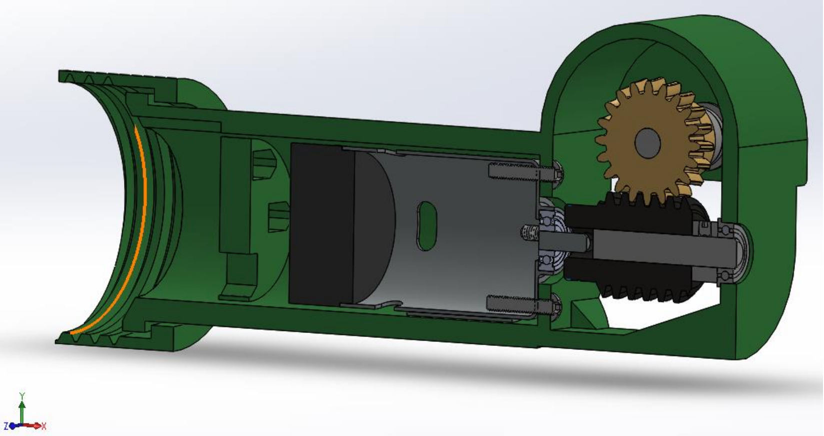


Figure 26 – Modeled Link with Female Port and Threaded Nut, Midsection View

The fittings for the XT60 Power Molex connectors, as referenced in the Bill of Materials, allow minimal tolerance around the perimeter of the connector. This is intended to serve a tight fit with minimal adhesive. Had this prototype been assembled, the intended adhesive was super glue since its low viscosity would allow a rigid bond around the entire profile of the Molex connector. In this design iteration, the electrical connections that need to travel between the male and female ports of the same link are exposed outside of the ABS-plastic housing. This aspect of the design could be improved in a future iteration to better protect the exposed electrical wires from potentially pinching at the link joint.

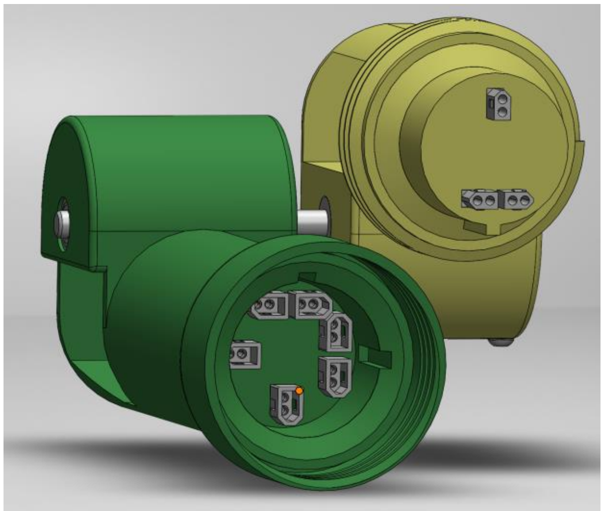


Figure 27 – Modeled Link with Female and Male Molex Connectors

Multiple Link Implementation

Given the budget and resource constraints of this project, the objective of the minimum viable product was to implement four links simultaneously as shown in Figure 28. It is essential to observe the varied angular configurations at which the four links are attached. In this particular four link configuration with four total degrees of freedom, all four three connection between the four links are offset by 90 degrees. In this model, the base joint and end effector are not shown.

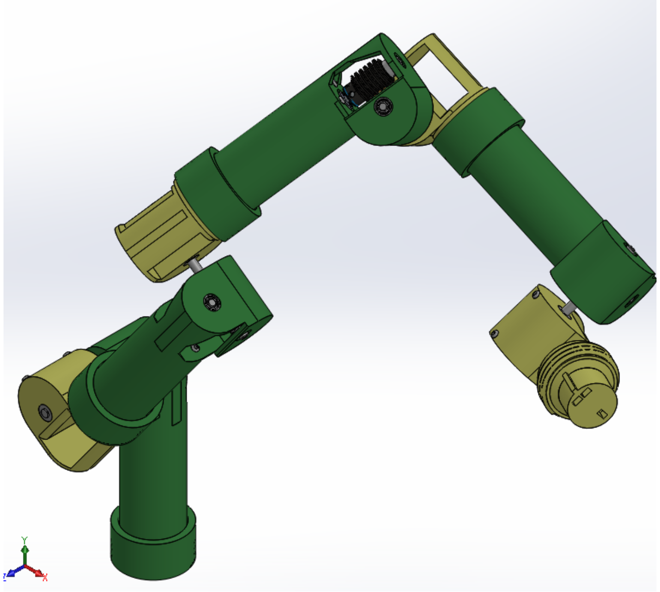


Figure 28 – Modeled Four Links Connected Sequentially in Varied Angular Configurations

Base Link

The base link is designed to be a stationary revolute joint for other links to attach to. The base link is intended to be mounted to a surface within workspace that is parallel to the floor. The joint is capable of full 360-degree rotation. Since the external moments exerted on the base link are about the X and Z axes due to gravity, the joint does not need to have a locking mechanism. Unlike the other links, the base link does not have a worm gear and worm wheel pair. The M4 size holes around the perimeter of the base allow the base to be rigidly mounted to a workbench operation. An isometric view of the modeled base link is shown in Figure 29.

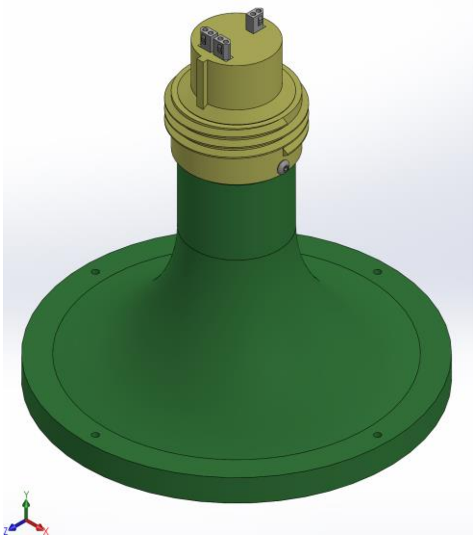


Figure 29 – Modeled Base Link – Isometric View

A midsection view of the base link assembly in shown in Figure 30. The top of the base link is a male port with the external threads to connect to another link. The external threads and male electrical ports are designed to operate just like any other link. The top half rests on a bearing connected to the motor housed within the stationary base. The bearing is implemented to withstand moments about the X and Z axes and reduce friction.

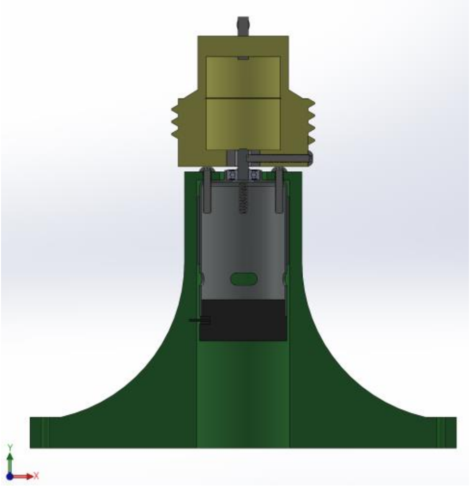


Figure 30 – Modeled Base Link – Midsection View

End Effector

The end effector needed to be simple and easy to troubleshoot. This is because the focus of the project was narrowed on the link platform design and its compatibility, not the end effector itself. As determined by the task specifications mentioned prior, the end effector in this project needed to be able to grasp a 1 kg payload. To achieve this, the open source design for a simple 1 degree of freedom claw was modified to include a female port. (Sethi, 2019) This approach was appealing because it used a small servo motor with lightweight 3D printed ABS plastic gears. The gripper was coupled with elastics to keep the gripper tightly grasped around the target payload. In addition, the gripper claw tips were modified to accommodate elastics grips to increase frictional forces when in contact with the payload surfaces. The gripper was modified to include the female port, including two sets of Molex connectors and the threaded ABS plastic nut. Since the connections match the rest of the platform synonymously, any link can be connected with the end effector. The end effector is modeled in Figure 31.

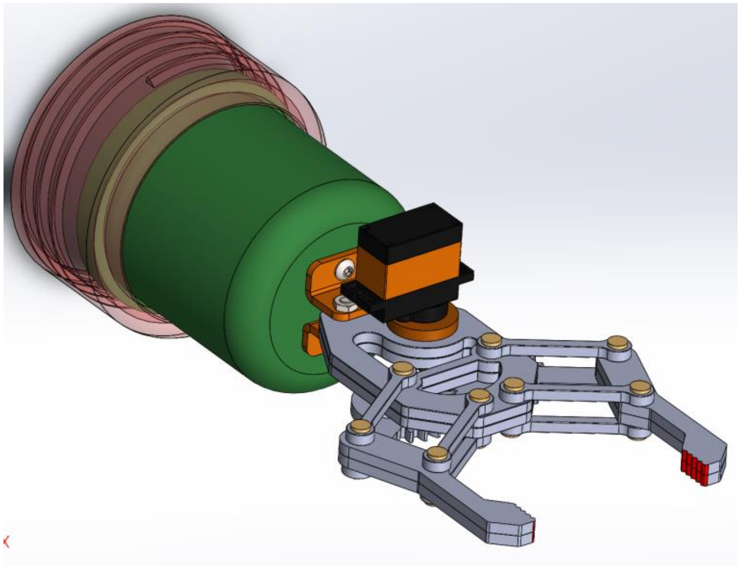


Figure 31 – Modeled Gripper w/ Female Port and ABS Threaded Nut

Configuration Planned for Demonstration

To conclude the features and versatility of the modular platform, the team planned a minimum viable product for physical demonstration at the Major Qualifying Project showcase. Due to the limitations of remote learning in D-term caused by COVID-19, the physical prototype was not able to be assembled and physically tested. This minimum viable product was planned as a 6 degree of freedom configuration with four links, the base link, and end effector. A SolidWorks assembly of this configuration is shown in Figure 32.

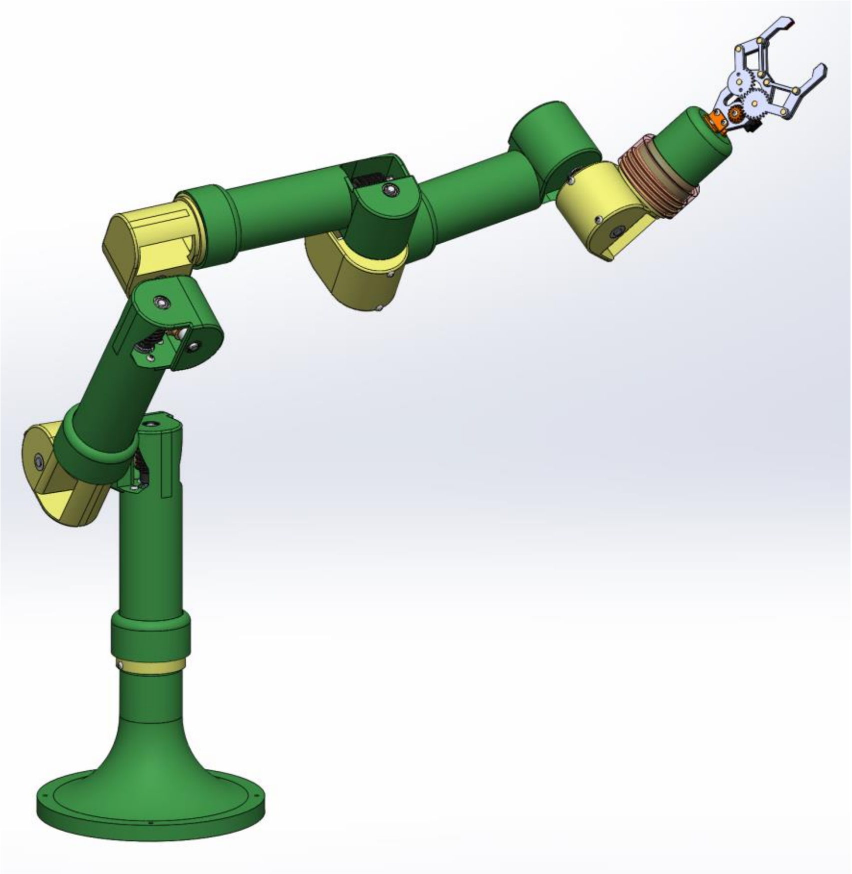


Figure 32 – Configuration planned for Demonstration with Four Links, Base Joint, and End Effector

Electronics Simulation

Current Draw of Motors

Each motor has a stall current of 13 amps and 6.5 amp optimal performance, and with the initial plan to use eight motors, the current demand would be quite large. Once the amount of motors was reduced to four, so was the task space, and current demand. Since the motors have the greatest current draw, it was best to simulate them operating using the motor driver and measure how much current the system will pull. Proteus Design Suite, a software tool used for electronic design automation, was used to simulate these motors connected as seen in Figure 33.

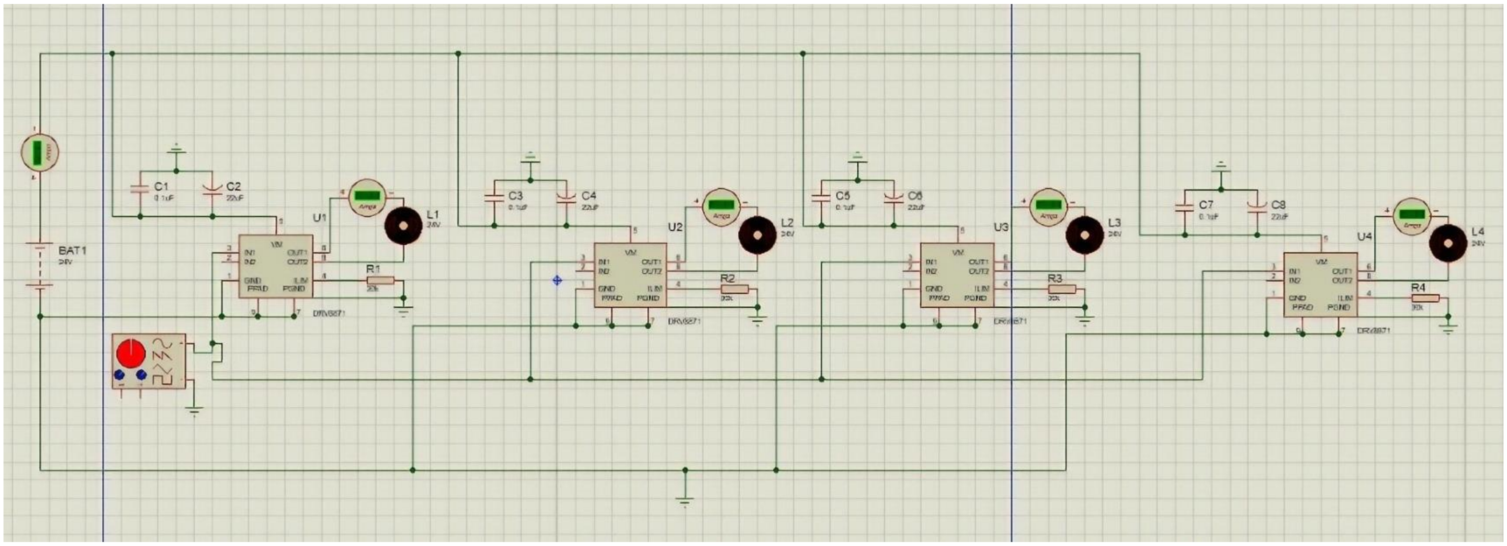


Figure 33 – Schematic of Motor Drivers and Motors

Figure 34 simulates a single motor and motor driver. Running this simulation did not just allow for better comprehension of the current characteristics in the modular arm, it also helped us decide which wire gauge would be necessary to implement and where protection may need to be put in place for the other electronics. The capacitors seen in the figure below represents the resistive load when the motor is turned on. As more motors are added to the system, the current demand to support them increases. Including additional motors is important to consider since this is an n-link robotic arm with the ability to support multiple joints. As the resistive load increases in each joint, the overall current through the arm decreases. This is good because it proves that the system is in parallel and that the current demand will not be as high in the additional joints after the first one.

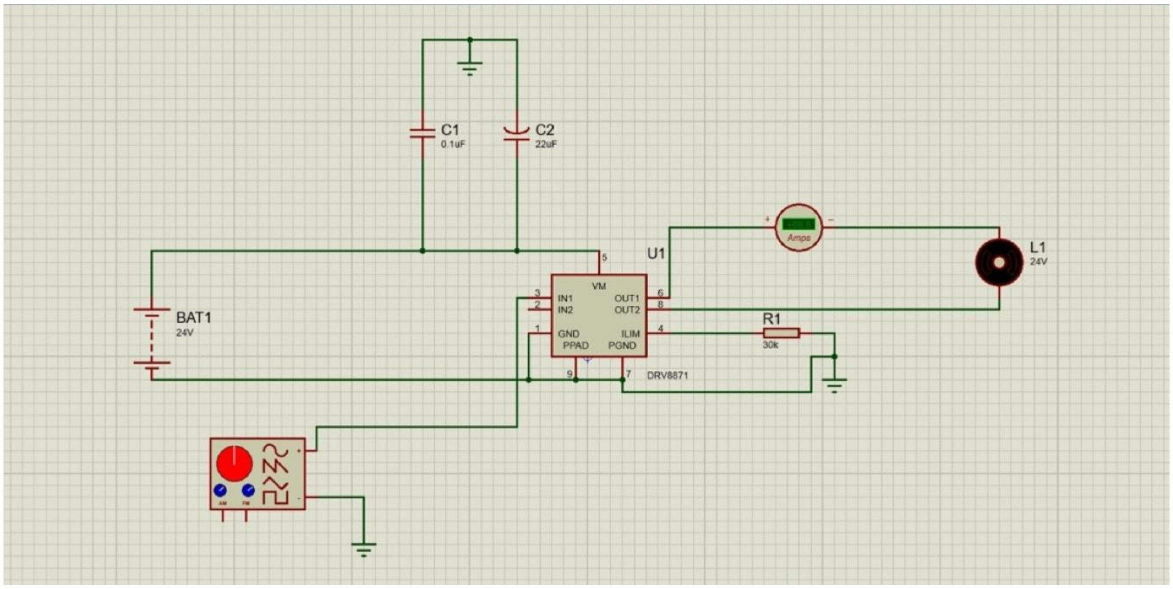


Figure 34 – Simulation for Current Draw in Motors

CAN Bus Terminating Communication

For the robot to complete specific configurations, data must be sent through a communication protocol instructing the robot to do so. When using a CAN Bus communication protocol for multiple links that are physically connected in series, the first and final link must be terminated with a 120-ohm resistor. Since the electronics will be housed internally within each of the links, the resistor cannot be terminated by hand, so a relay was introduced to electronically switch the resistors on and off as links are attached and removed. Using Proteus Design Suite, the functionality of the relays in the robot terminating the resistors were demonstrated, as seen in Figure 35. The SPDT relay start normally closed and become open once the robot is powered on. The switch connected to the relay, being pressed, represents each link being attached, when not pressed, it is the final link of the system. The resistor in the first link always needs to be terminated, so it should always be connected to the closed terminal. When the robot is turned on the two relays in the middle disconnect from the resistor allowing data to be transferred, and because nothing is connected to the final link, the switch is not pressed, and the relay remains closed which terminates the final link.

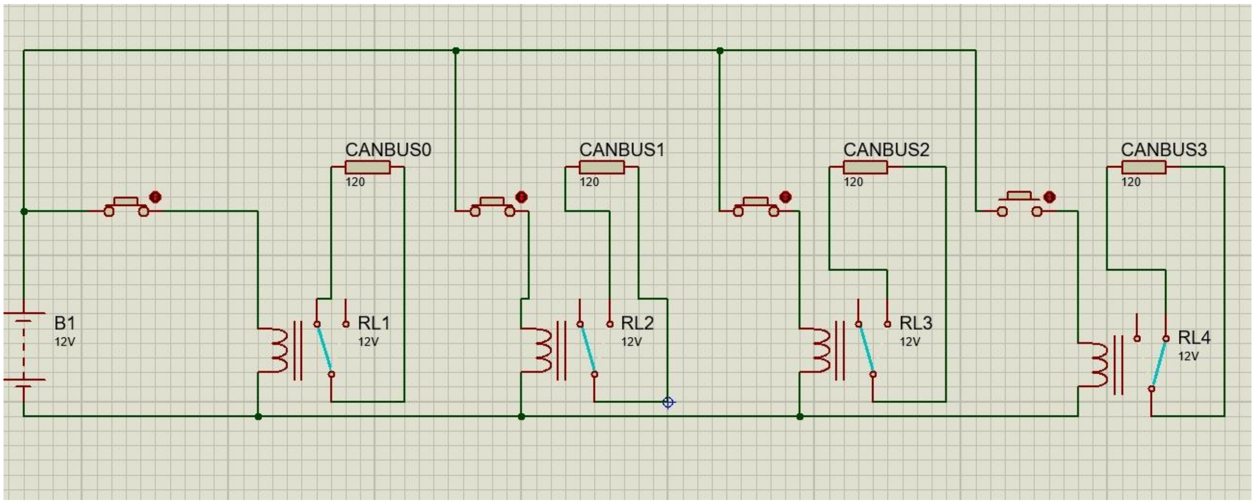


Figure 35 – Simulation of Relays Termination Resistors

Full Link Simulation

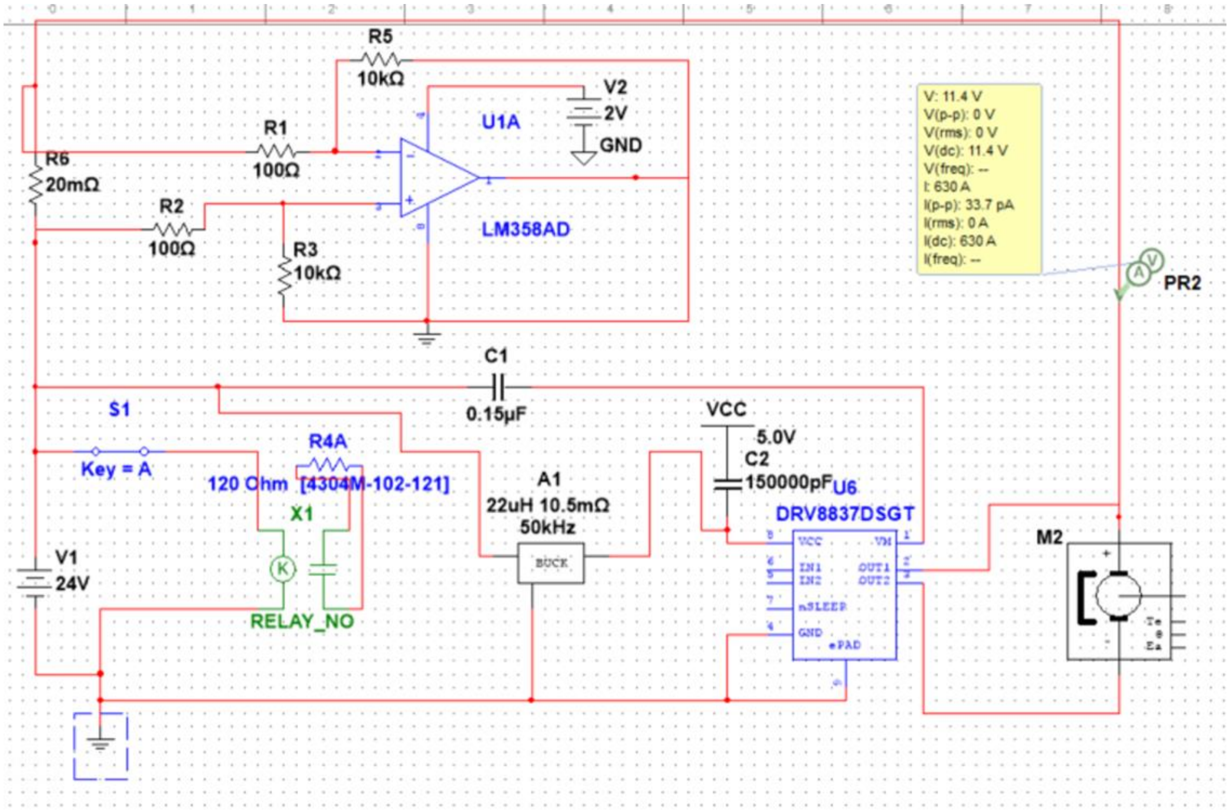


Figure 36 - Base Link Simulation in Multisim

The previous schematics seen in figure 28 and figure 29 demonstrate specific characteristics that were a primary focus when developing the electronics for this modular robot. The schematic shown in figure 35 represent the electronics that are connected at the base (shoulder joint) and the first link of the arm. This schematic was designed using NI Multisim, an industry-standard SPICE simulation and circuit design software for digital, analog, and power electronics. These components will be in each of the links except for the current sensor, which will only be attached at the base.

PCB Design

Once the schematic was designed to test the functionality of the electrical components selected for each link, the parts were created in EAGLE. Autodesk EAGLE is an electronic design automation (EDA) software that allows printed circuit boards (PCBs) to easily connect to schematic diagrams, place components, PCB routing assistance, and an extensive part library. The space available in each link to enclose the electronics is very limited, designing a PCB will minimize the space that is taken up by wires and sporadic space. First, the schematic had to be recreated in EAGLE. For the more specific parts like the ATMEGA32U4 Microcontroller and the MCP2515 CANbus Controller, SnapEDA was used to import the product symbol for the schematic and footprint for designing the PCB. SnapEDA is an electronics design library that assist engineers in building products faster by moving design barriers. Although the electronics were tested in in a singular system on Multisim, the actual design is split into two separate PCBs. The design in figure 37 holds a motor driver, relay, switch, terminating resistor, and buck converter that will all be in the PCB directly behind the motor. The design in figure 38 houses the CANbus module, microcontroller, relay, switch, and terminating resistor. The green lines represent the net where the traces will go on the PCB so that when the components are soldered on later, they have a connection through the coper trace. The nets that cut off, have labels that represent where the wires connect to the other PCB via data or power lines.

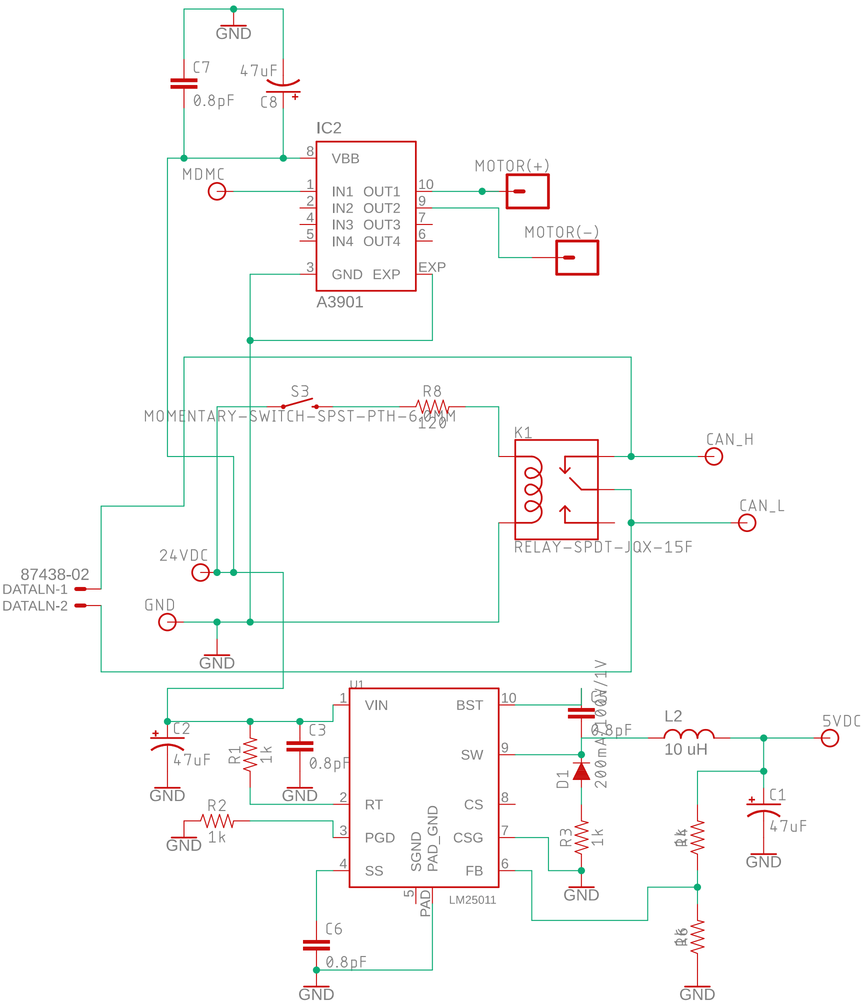


Figure 37- Schematic for PCB Behind Motor

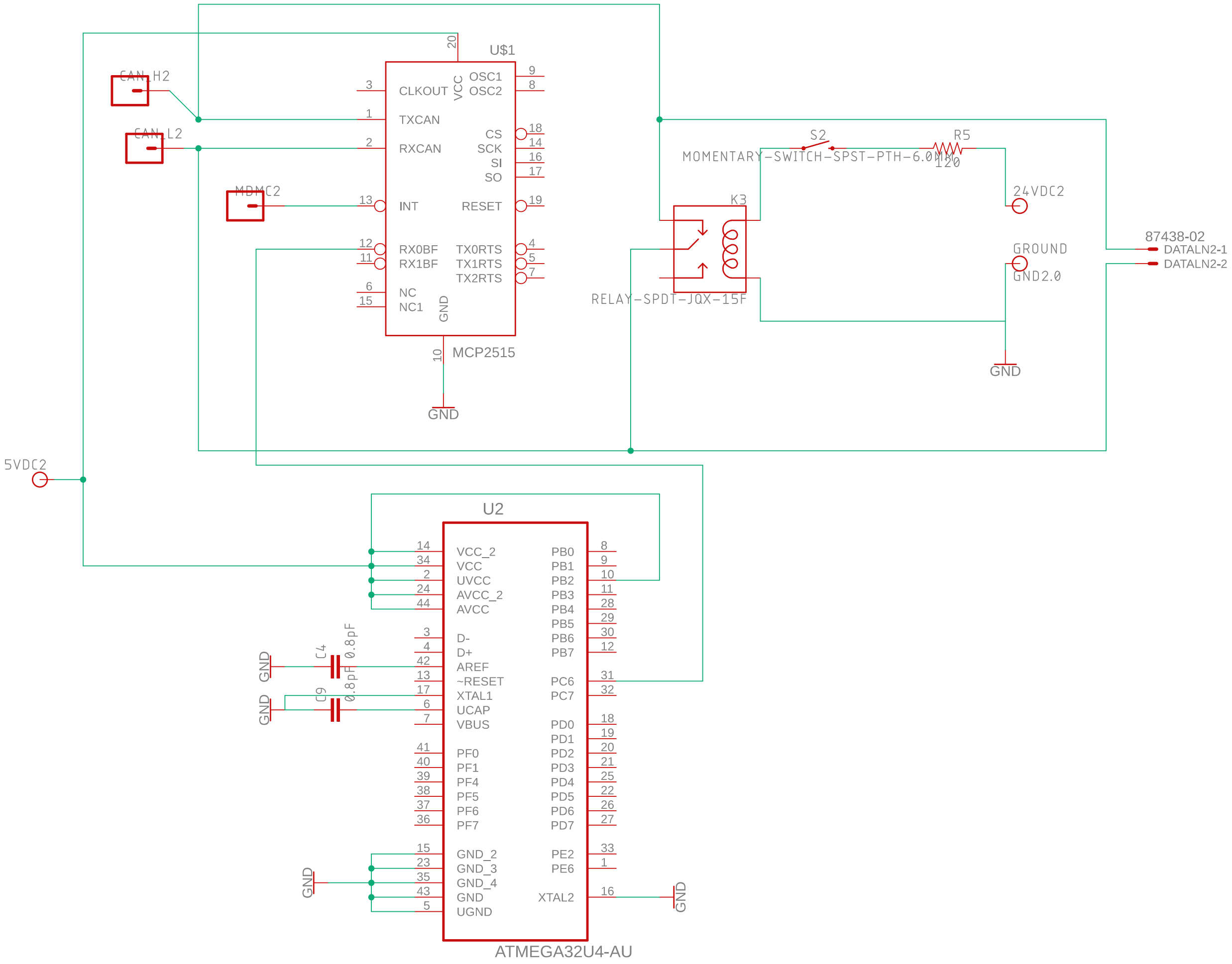


Figure 38- Schematic for PCB Behind Molex Connector

Once the schematic was completed, the footprints and nets were transferred over to the board to be positioned for printing. Typically, the canvas is a square but this one has been modified to fit the diameter 45 mm of the interior for each link. Ground poly, the blue circle running parallel to the PCB, was added to reduce the number of traces required since anything connected to ground will connect to this bottom side (Layer 16). The placement was done based on the component with the most threads that needed to be traced because the traces cannot overlap or else it will disrupt the connection. The largest component was placed, and the similar components were grouped and placed accordingly. The net wires that were initially yellow on the board (untraced) turned to red once they were traced and connected to each other. The trace thickness was set to 12 since these PCBs are dealing with a higher current. The wires will run adjacent to the links and go inside of the link through a small incision where the PCBs are located. Both PCBs contain holes that will allow the power, ground, and data lines to go through so that they can connect to the Molex connectors and the electrical connection is made between each link.

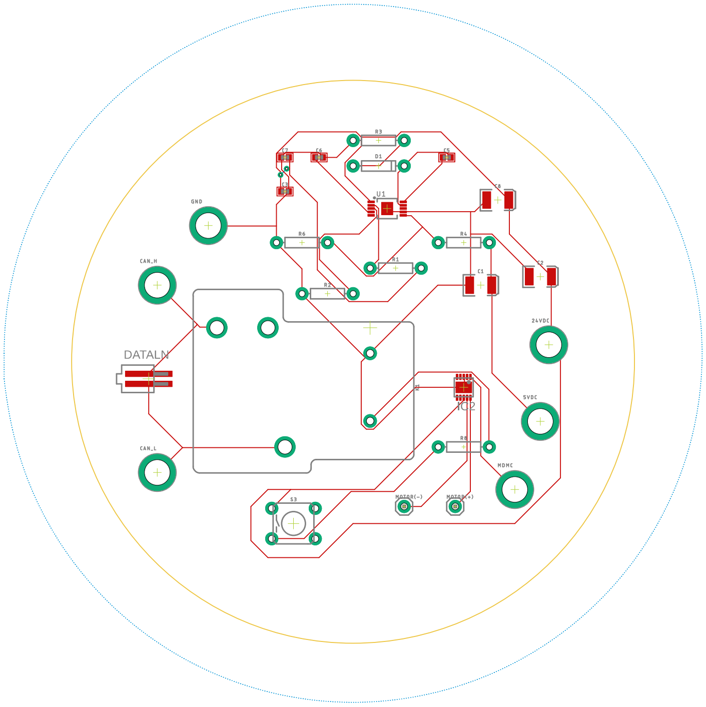


Figure 39 - PCB Behind Motor

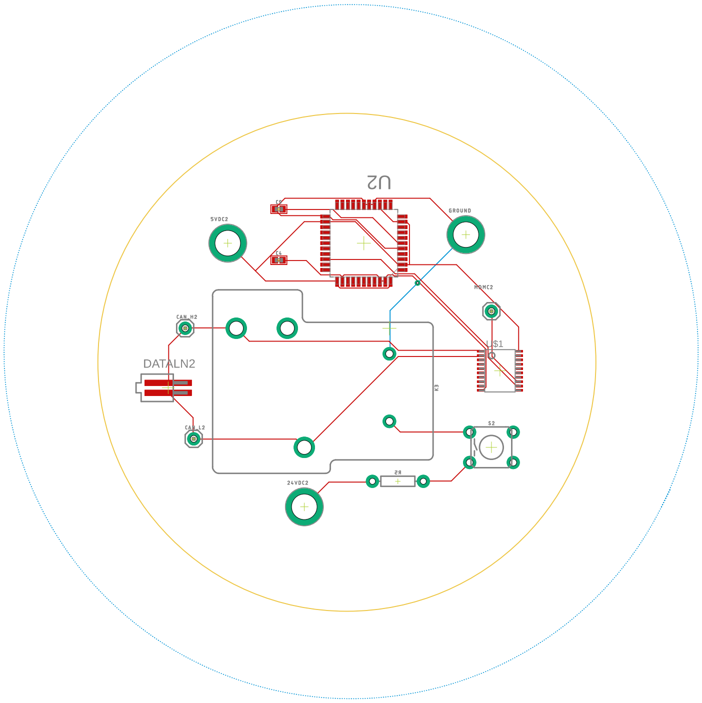


Figure 40 - PCB Behind Molex Connector

Conclusion

The goal of this project was to develop an N-link robotic arm that was capable of dynamically positioning its end effector in a specified task space based on the number of links that were attached. In addition, it was desired to have a fault-tolerant system that could handle adding and removing links without requiring a reboot. This platform would allow a single link to be manufactured in mass quantities to serve a wide variety of configurable applications. Theoretically, mass production of this modular link platform would allow for a business model with greater profit margins. Although the prototype was not able to be physically constructed due to the COVID-19 pandemic, the evaluation of digitally testing the parameters of the robot verifies how the robot would function if it were built. If the project were completed on campus, it would have contained 3D printed joints and base link, with the electronics housed inside, and communication between the links would be visible as the robot would complete a task with its end-effector at various lengths.

Through this project, the ability to adapt to abrupt project changes was a great learning point, as well as communicating frequently about the project design and progress. This project required the interdisciplinary background of mechanical engineering, electrical engineering, and computer science. Considering the diverse backgrounds involved in this project, and pushing the boundaries of research and technical expertise, great lead way was made in developing a product to elevate the modular robotics industry.

By the year 2021, the International Federation of Robotics estimates that the number of operational industrial robots will increase to 3,788,000. With the largest industry consumer being the automotive industry with a 30% market share, the implementation of this modular robotic arm would be extremely efficient in reducing downtime, maneuvering large and small parts, and improving manufacture speed which will ultimately cut down company manufacturing cost.

Since the development of this technology is quite relevant to this continuous growth of the industrial robotics industry, there is great potential for this concept. Our team hopes that this concept will continue to be improved in the future. First, we suggest testing the physical system as provided by our designs. The printed circuit boards should be produced and evaluated to make the link housing more compacted. It is instrumental to compare the quantitative performance data that is collected with our task specifications. Regarding the mechanical design, a thorough strength analysis should be completed to truly understand the limits of 3D printing these mechanical components. This may prompt design modifications to manufacture the parts with plastic molding to improve structural rigidity. If budgeting is not constrained, negotiate an academic partnership with Harmonic Drive or another OEM manufacturer for a motor with a higher torque to mass ratio. After considering the performance of these aspects of the design, modifications to the current iteration can be made. Software improvements could be made to the control algorithm of the arm. The current controls simulation uses a simple joint space based PID with gravity compensation. This control loop could be improved to provide task space control, which would provide more stable end effector motion.

Appendix

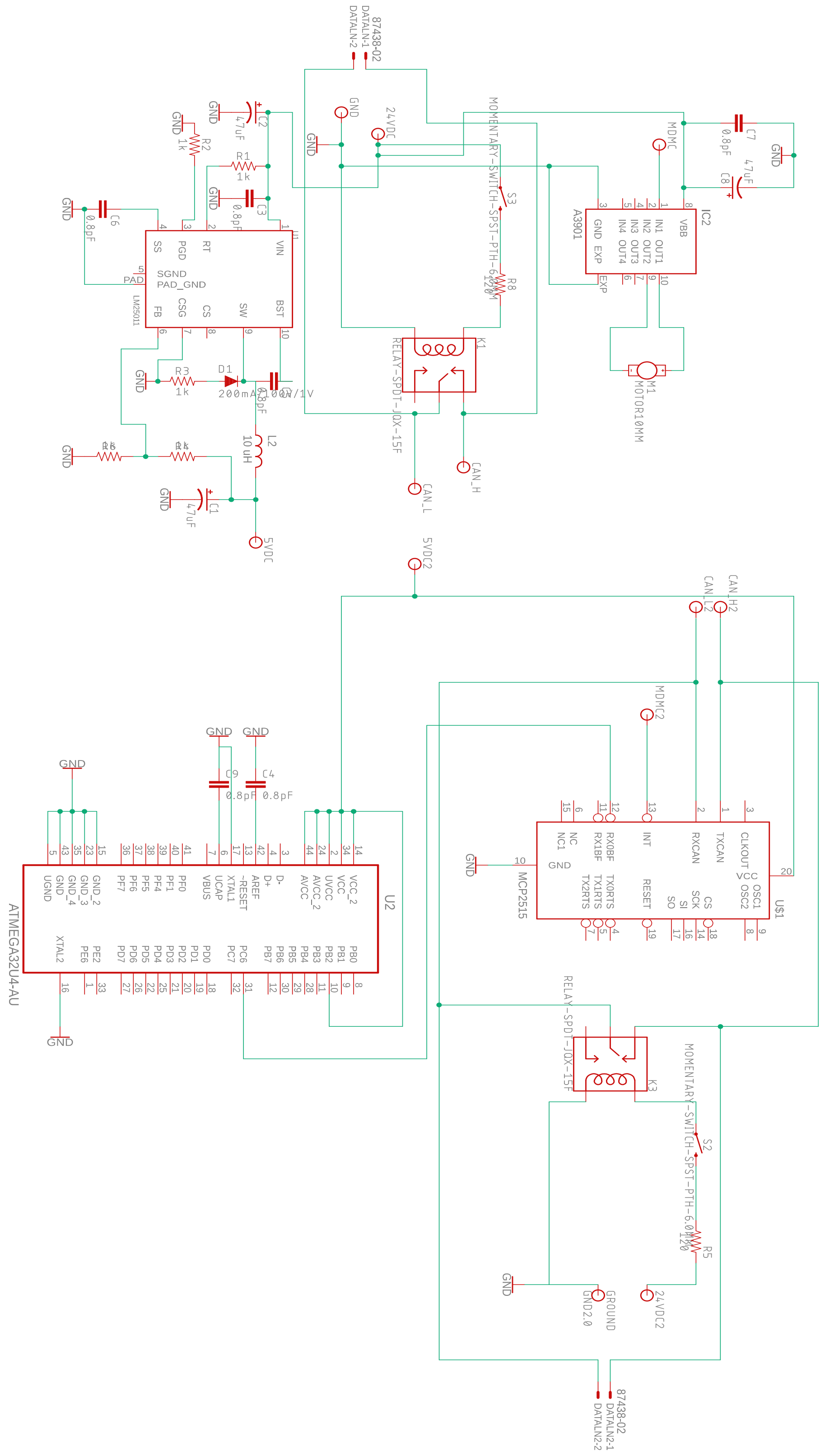
Bill of Materials

The following bill of materials includes all parts as designed or purchased to assemble a single link. Since this platform is a modular system, how many links are needed for a particular configuration must be considered.

#	Part ID	Description	Qty.
1	Devantech 24V	49:1 Gear Motor w/ Encoder; https://www.robotshop.com/en/devantech-24v-49-1-gear-motor-encoder.html Open source CAD: https://grabcad.com/library/emg49-dc-motor-1	1
2	57485K67	Shaft Collar w/ Set Screw - Black-Oxide 1215 Carbon Steel; https://www.mcmaster.com/57485K67	4
3	7804K147	Flanged Ball Bearing – Shielded Stainless Steel; https://www.mcmaster.com/7804k147	3
4	Housing - Female End	Female end of joint housing composed of 3D printed ABS plastic [Part10.sldpt]	1
5	KHK US-SW1 5-R1J8	Module 1.5, Single Lead, Right Hand, Carbon Steel Worms; https://www.khkgears.us/catalog/product/SW1.5-R1J8	1
6	KHK S-BG1 5-20R1J8	Module 1.5, 20 Tooth, Single Start, Right Hand, Bronze Worm Wheel; https://www.khkgears.us/catalog/product/BG1.5-20R1J8	1
7	92095A196	M4 Button Head Hex Drive Screw - Stainless Steel 0.70 mm Thread, 20mm Long; https://www.mcmaster.com/92095a198	4
8	1265K64	8mm Stainless Steel Rotary Shaft (200mm long per unit); https://www.mcmaster.com/1265k64	1
9	2011N112	Oil-Embedded Thrust Bronze Bearing; https://www.mcmaster.com/2011n112	1
10	Housing – Male End	Male end of joint housing composed of 3D printed ABS plastic [link_end.sldpt]	1
12	92095A198	M4 Button Head Hex Drive Screw - Stainless Steel 0.70 mm Thread, 30mm Long; https://www.mcmaster.com/92095a198	2
13	3061023	Male Molex Pin & Socket Connector https://www.molex.com/molex/products/part-detail/crimp_housings/0003061023	3
14	3062023	Female Molex Pin & Socket Connector https://www.molex.com/molex/products/part-detail/crimp_housings/0003062023	6

15	Anmbest_MD114	https://www.amazon.com/Anmbest-High-Power-Adjustment-Electronic-Brightness/dp/B07NWD8W26/ref=pd_di_sccai_33?_encoding=UTF8&pd_rd_i=B07NWD8W26&pd_rd_r=1c017f48-d3bb-402c-970c-e0b3955b87dd&pd_rd_w=qvHdb&pd_rd_wg=ZiG17&pf_rd_p=e532f109-986a-4c2d-85fc-16555146f6b4&pf_rd_r=XH16G897Z0PTKWSJZ1S4&psc=1&refRID=XH16G897Z0PTKWSJZ1S4	1
16	4226	https://www.adafruit.com/product/4226?gclid=CjwKCAiAyeTxBRBvEiwAuM8dnarETh2yc8J9Mu8OpxjjSmuBEhJuAyf-g2mCtZT0IwBGIGjR6iZ_tRoCIPoQAvD_BwE	1
17	LT1766EFE-5#PBF-ND	https://www.digikey.com/product-detail/en/linear-technology-analog-devices/LT1766EFE-5-PBF/LT1766EFE-5-PBF-ND/961248	1
18	Z990-ND	https://www.digikey.com/product-detail/en/omron-electronics-inc-emc-div/G6RN-1-DC24/Z990-ND/260144	1
19	Wire-RB-12-50	https://powerwerx.com/red-black-bonded-zip-cord?gclid=CjwKCAjwqdn1BRBREiwAEbZcR6-ZaLWTAK7LnylLmtZ0Av5Uy70zZAAr-_GZzq2vNollA98wBnDkWhoCFmcQAvD_BwE	1

Schematic Designed for PCB on EAGLE



Bibliography

Bolton, C., Cantos, M., & Tang, P. (2015). *Real-time Control of Robot Arm Based on Hand Tracking Using Leap Motion Sensor Technology*. (Undergraduate Major Qualifying Project No. E-project-012415-140406) From Worcester Polytechnic Institute Electronic Project Collection: https://web.wpi.edu/Pubs/E-project/Available/E-project-012415-140406/

Burridge, R. (2011). *Projects: RMM*. From TRACKLabs: https://traclabs.com/projects/rmm/

Caccavale, F. (1997). Second-Order Kinematic Control of Robot Manipulators with Jacobian Damped Least Squares Inverse. *IEEE ASME Transactions on Mechatronics*, 191.

Campeau-Lecours, A., Lamontagne, H., Latour, S., Fauteux, P., Maheu, V., Boucher, F., . . . L'Ecuyer, L.-J. (2017). Kinova Modular Robot Arms for Service Robotics Applications. *International Journal of Robotics Applications and Technologies, 5*, 49-71. doi:10.4018/IJRAT.2017070104

Corrigan, S. (August 2002 - Revised May 2016). *Introduction to the Controller Area Network (CAN).* Texas Instruments. Texas Instruments.

Dhaker, P. (2018, September). Introduction to SPI Interface. *Analog Dialouge*(52-09). From Analog.

Di Vito, D., Natale, C., & Antonelli, G. (2017). A Comparison of Damped Least Squares Algorithms for Inverse Kinematics of Robot Manipulators. *International Federation of Automatic Control*.

English, J., & Maciejewski, A. (1998). Fault Tolerance for Kinematically Redundant Manipulators: Anticipating Free-Swinging Joint Failures. *IEEE Transactions on Robotics and Automation, 14*(4), 566-575. doi:10.1109/70.704223

Giovino, B. (2010). *Application & Technologies: Sensors*. From Mouser Electronics: https://www.mouser.com/applications/making-sense-current-sensing/

Hopkins, M. (2018). *Blisk Inspection System*. (Undergraduate Major Qualifying Project No. E-project-012218-213509) From Worcester Polytechnic Institute Electronic Project Collection: https://web.wpi.edu/Pubs/E-project/Available/E-project-012218-213509/

I2C Bus, Interface, and Protocol. (2020, January). From I2C Info: https://i2c.info

Jia, T., Liu, Y., & Li, J. (2019). Fault Detection for Mechanical Arm Systems: An Sliding Mode Observer Approach. *2019 IEEE 3rd Information Technology, Networking, Electronic, and Automation Control Conference*, 1541-1544. doi:10.1109/ITNEC.2019.8729344

Jiang, Z., Gao, W., & Yu, X. (2018). An Innovative Error Measuring Method for Modular Interfaces of Modular Reconfigurable Robots. *Proceedings of the 3rd International Conference on Robotics, Control and Automation, ICRCA '18*, 119-122. doi:10.1145/3265639.3265674

Lewis, C., & Maciejewski, A. (1997). Fault Tolerant Operation of Kinematically Redundant Manipulators for Locked Joint Failures. *IEEE Transactions on Robotics and Automation, 13*(4), 622-629. doi:10.1109/70.611335

Maloney, T., & Topper, A. (2019). *Piano-Playing Robotic Arm*. (Undergraduate Major Qualifying Project No. E-project-042919-161531) From Worcester Polytechnic Institute Electronic Project Collection: https://web.wpi.edu/Pubs/E-project/Available/E-project-042919-161531/

O'Shea, C., Taglieri, A., & Titus, B. (2018). *MIRA: Modular Interchangeable Robotic Arm*. (Undergraduate Major Qualifying Project No. E-project-042518-184949) From Worcester Polytechnic Institute Electronic Project Collection: https://web.wpi.edu/Pubs/E-project/Available/E-project-042518-184949/

Prabhala, V. A. (2018). An Overview of Direct Current Distribution System Architectures & Benefits. *Energies*.

Prabhala, V., Baddipadiga, B., Fajri, P., & Ferdowski, M. (2018, September). An Overview of Direct Current Distribution System Architectures & Benefits. *Energies, 11*, 2463. doi:10.3390/en11092463

Schrock, P., Farelo, F., Alqasemi, R., & Dubey, R. (2009). Design, simulation and testing of a new modular wheelchair mounted robotic arm to perform activities of daily living. *2009 IEEE International Conference on Rehabilitation Robotics*, 518-523. doi:10.1109/ICORR.2009.5209469

Ting, Y., Tosunoglu, S., & Tesar, D. (1993). A control structure for fault-tolerant operation of robotic manipulators. *[1993] Proceedings IEEE International Conference on Robotics and Automation, 3*, 684-690. doi:10.1109/ROBOT.1993.291826

Zhou, Q., Liu, M., Wang, W., & Xu, L. (2018). Research on Fault Tolerant Control Strategy of Multi-Degree-of-Freedom Manipulator with Single Joint Faults. *Proceedings of the 37th Chinese Control Conference*, 5853-5859. doi:10.23919/ChiCC.2018.8483976

Ziegler, S., Woodward, R. C., Iu, H. H.-C., & Borle, L. J. (2009, April). Current Sensing Techniques: A Review. *IEEE Sensors Journal, 9*, 354-376. doi:10.1109/JSEN.2009.2013914